FACULTAD DE ESTUDIOS ESTADÍSTICOS

Máster de Minería de Datos e Inteligencia de Negocios.

Curso 2016/2017

Trabajo de Fin de Máster

TITULO: Estudio de la competencia de los precios de los carburantes en España mediante Teoría de Juegos y Machine Learning.

***Alumno*: Beatriz Escribano Pablos.**

***Tutor*: Javier Castro Cantalejo.**

Noviembre de 2017

UNIVERSIDAD COMPLUTENSE
MADRID

Índice

Índice de figuras

Índice de tablas

1 Introducción

En la actualidad, el crecimiento exponencial de Internet genera grandes volúmenes de información relevante que es necesario buscar relaciones y patrones. Este nuevo paradigma se conoce como Minería de datos, la cual se basa en la metodología SEMMA: Sample, Explore, Modify, Model, Asses.

Para la creación de modelos, las técnicas están basadas en Machine Learning, que consiste en diseñar e implementar algoritmos que permitan "aprender" a un ordenador a resolver problemas de forma automática, generando soluciones del problema, a partir de unos datos de prueba, y realimentarse para obtener mejores soluciones al problema [33].

En 1960, los pocos automóviles que circulaban por las ciudades españolas se abastecían de gasolina en los surtidores o bombas de gasolina instalados cerca de la estación del ferrocarril en el Progreso y en Sáez Díez. Funcionaban a mano con un sistema de manubrio y expedían 5 litros. En los años 80, el aumento del número de coches hizo necesario un cambio en el modo del funcionamiento y de la localización, lo que propició el auge de las estaciones de servicio en las carreteras [1].

En 2015, según la Asociación Española de Operadores de Productos Petrolíferos (AOP), el número de gasolineras creció un 20.3% desde el inicio de la crisis en 2007 [2].

Dada la importancia de este sector, en un mundo cada vez más globalizado, en el presente estudio nos centraremos en distintas técnicas de Machine Learning para predecir el precio del carburante de las gasolineras de España, y las compararemos para concluir cuál de esos modelos es el que mejor predice con nuestra base de datos.

Además, vamos a ofrecer otra forma de obtener predicciones, basándonos en teoría de juegos. La teoría de juegos es la ciencia estratégica que analiza las interacciones de los individuos ante situaciones en las que las decisiones de una o más personas pueden influir en las decisiones y en la utilidad de otras personas, teniendo en cuenta las reacciones de los demás participantes [34], la cual se utiliza tanto en la economía, que será nuestro caso, como en la vida misma a la hora de decidir el pago fraccionario o no de la comida en un restaurante con tus amigos.

La memoria está estructurada de la siguiente forma:

- En el Capítulo 2 se explica brevemente los distintos tipos de gasolina para hacer más fácil el presente estudio.
- En el Capítulo 3, se muestran los datos a analizar en este estudio.
- En el Capítulo 4, se establecen los objetivos fundamentales de este.
- En el Capítulo 5, se explica la metodología estadística que se ha utilizado para llevar a cabo los objetivos.

- En el Capítulo 6, se realiza un análisis descriptivo de las variables de una gasolinera, ya sea su ubicación, como el tipo de gasolina, además de las variables de la competencia, referidas al precio de aquellas gasolineras que se encuentran a una determinada distancia.

- En el Capítulo 7, se lleva a cabo la depuración de los datos, como, la recodificación de las categorías de alguna de las variables, y el análisis de missing y atípicos.
- En el Capítulo 8, se explican las variables finales que van a ser objeto de estudio de los puntos siguientes.

- En el Capítulo 9, se lleva a cabo un análisis multivariante para establecer relaciones entre las variables y así, poder tener un mayor conocimiento del presente estudio.

- En el Capítulo 10, se realizan distintos modelos predictivos de Machine Learning, para predecir el precio del carburante de las gasolinas de España.

- En el Capítulo 11, se obtiene el mejor modelo de predicción de los realizados en el capítulo anterior.

- En el Capítulo 12, se lleva a cabo un análisis de competencia empresarial en el que se predecirá el precio de las gasolineras de una determinada zona de España, teniendo en cuenta la competencia, basándonos en teoría de juegos, más concretamente, el Juego de Bertrand.

- En el Capítulo 13, se presentan las conclusiones y el trabajo futuro.

2 Explicación de los tipos de gasolina

En este apartado, vamos a explicar de forma breve cada uno de los tipos de gasolina que son objeto de estudio.

Los tipos de gasolina son los siguientes:

2.1 Gasóleo A

Gasóleo A [24] se caracteriza por:

- Es el gasóleo de más “calidad”.
- Adecuado para vehículos de automoción. Está más refinado y contiene aditivos para evitar la solidificación de la parafina a bajas temperaturas y que además aportan una serie de beneficios para el vehículo como son, por ejemplo, reducir el consumo y las emisiones contaminantes, y proteger la bomba y el sistema de inyección.

2.2 Gasóleo B

Gasóleo B [24] se caracteriza por:

- Es el gasoil que se usa para maquinaria agrícola, pesquera, embarcaciones y vehículos autorizados.
- Está menos filtrado y contiene más parafina que el gasóleo A, con lo que puede generar problemas en el mantenimiento en coches y motos.
- Su uso fuera del ámbito indicado está considerado como un delito de fraude o estafa a la Hacienda Pública ya que se estarían evitando los impuestos estatales a pagar si se tratase de gasóleo A.

2.3 Gasolina 95 Protección

Este tipo de gasolina [23] se caracteriza por:

- Es para preservar los motores de automóviles veteranos.
- El combustible tiene un contenido máximo de oxígeno de 2,7 por ciento en masa y un contenido máximo de etanol de 5 por ciento en volumen.
- Contenidos mayores podrían hacer que motores anteriores al año 2000 se estropearan.

2.4 Gasolina 98

Gasolina 98 [22] se caracteriza por lo siguiente:

- Es una gasolina con menos contenido en azufre (incluso menos que la de 95), y, en el caso de algún producto, totalmente libre de este compuesto. Se convierte así en uno de los combustibles más limpios.
- Su composición tiene nuevos aditivos de última generación.
- Mejora las prestaciones del motor, su protección y disminuye el consumo.
- Alarga la vida útil del catalizador.
- Optimiza las prestaciones de vehículos de gama alta.

2.5 Biodiésel

Este tipo de gasolina [28] se caracteriza por:

- Es un líquido que se obtiene a partir de lípidos naturales como aceites vegetales o grasas animales, mediante procesos industriales de esterificación y transesterificación.

2.6 Bioetanol

El bioetanol [29] es

- Un combustible en el que las bacterias son las responsables de degradar estos residuos y de producir este elemento.
- Se utiliza en motores de explosión como aditivo o sustitutivo de la gasolina.

2.7 Gases Licuados Petróleo (GLP)

Los Gases Licuados Petróleo [26] son:

- Una mezcla entre propano y butano disueltos en petróleo y obtenidos durante el refinado de éste.
- Son líquidos.
- Contamina mucho menos, ya que un vehículo funcionando con GLP emite un 15% menos de CO_2 y entre un 70 y un 90% menos de óxidos de nitrógeno.

2.8 Gas Natural Comprimido (GNC)

El Gas Natural Comprimido [27] se trata de:

- Es gas natural almacenado a altas presiones. Este gas natural es principalmente metano, que al tener un alto índice de hidrógeno por carbono produce menos dióxido de carbono.
- El gran problema de este combustible es que en España no es fácil encontrar un lugar donde lo vendan, a pesar de que vehículos como el Seat Mii Ecofuel o el Audi A3 Sportback g-Tron lo usan.

2.9 Gas Natural Licuado (GNL)

El Gas Natural Licuado [26] se caracteriza por:

- Es gas natural, pero tratado de tal manera que adopta una forma líquida.
- Es la mejor alternativa para monetizar reservas remotas y aisladas, donde no es económico llevar el gas al mercado directamente.
- Se somete a un proceso que incluye reducir su volumen hasta 600 veces y mantenerlo a la presión de -160 grados centígrados, lo que permite un almacenamiento más rápido, cómodo para su transporte.
- Es inodoro, incoloro y no tóxico.
- Es respetuoso con el medio ambiente ya que reduce las emisiones de óxidos de azufre, óxidos de nitrógeno, dióxido de carbono.

3 Naturaleza de los datos

Los datos del presente estudio recogen información temporal sobre las 10000 gasolineras españolas registradas en la página web Geoportal Gasolinera [22]. La temporalidad de los datos es diaria, siendo los primeros datos del día 11 de marzo de 2017 y los últimos del 2

de mayo del mismo año, es decir, se tiene información de un total de 2 meses, aproximadamente. Consta de 42178789 observaciones y 13 variables, las cuales son:

1) Fecha
2) Dirección
3) Código postal
4) Horario
5) Latitud
6) Longitud
7) Localidad
8) Margen: posición en la que está situada la gasolinera (D: derecha, I: izquierda, N: a ambos lados)
9) Municipio
10) Rótulo: nombre de la gasolinera
11) Provincia
12) Precio
13) Tipo gasolina (Gasóleo A, Gasóleo B, Gasolina 95 Protección, Gasolina 98, Biodiésel, Bioetanol, Gases Licuados Petróleo, Gas Natural Comprimido, Gas Natural Licuado)

Una vez descargados los datos de partida, se precisó de una fase de limpieza de datos, entre los que destacan, la eliminación de datos no válidos que no contenían los valores de dicha variable. Esto se realizó principalmente para la variable "precio" ya que contenía valores que no eran numéricos. Además, tuvimos que quedarnos solo con los datos referentes a un día y a una gasolinera, ya que, para cada día, hora y gasolinera, teníamos distintos datos, lo que implicó que cambiásemos el formato de la variable "fecha" en "día/mes/año" (en la página web también disponíamos de hora y minutos, es decir, teníamos el siguiente formato "día/mes/año hora/minutos", por ejemplo, "2017-04-04T07:54Z"), y que además, tuviésemos que recalcular los datos de nuestra variable "precio", como el promedio de los precios de un mismo día, la cual la denominamos "media_precio". Consta de 1315464 observaciones.

Un subconjunto de los datos depurados se puede observar en la Figura 8.

	fecha	direccion	cp	horario	latitud	longitud	localidad	margen	municipio	rotulo	provincia	media_precio	tipo_gasol
1	2017-03-11	420 km 275	13640	L-D: 06:00-12:00	39,37325	-3,344889	HERENCIA	D	Herencia	PETRONOR	CIUDAD REAL	1,184	gasoleoA
2	2017-03-11	420 km 275	13640	L-D: 06:00-12:00	39,37325	-3,344889	HERENCIA	D	Herencia	PETRONOR	CIUDAD REAL	1,259	gasolina95Proteccion
3	2017-03-11	420 km 275	13640	L-D: 06:00-12:00	39,37325	-3,344889	HERENCIA	D	Herencia	PETRONOR	CIUDAD REAL	1,399	gasolina98
4	2017-03-11	426 km 1	29120	L-D: 24H	36,652889	-4,679056	ALHAURIN EL GRANDE	D	Alhaur<ed>n el Grande	CAMPSA	M<c1>LAGA	1,219	gasoleoA
5	2017-03-11	426 km 1	29120	L-D: 24H	36,652889	-4,679056	ALHAURIN EL GRANDE	D	Alhaur<ed>n el Grande	CAMPSA	M<c1>LAGA	1,309	gasolina95Proteccion
6	2017-03-11	426 km 1	29120	L-D: 24H	36,652889	-4,679056	ALHAURIN EL GRANDE	D	Alhaur<ed>n el Grande	CAMPSA	M<c1>LAGA	1,449	gasolina98

Figura 8: Subconjunto de los datos de precios de los carburantes, "precios_gasol_def"

```
> summary(precios_gasol_def)
     fecha                          direccion          cp                horario          latitud          longitud
 Min.   :2017-03-11   AVENIDA ANDALUCIA, S/N       :  1139   04700  :  3504   L-D: 24H       :581040   Min.   :27.75   Min.   :-18.0119
 1st Qu.:2017-03-23   AVENIDA JUAN CARLOS I, S/N   :   852   30500  :  3128   L-D: 06:00-22:00:211020   1st Qu.:38.05   1st Qu.: -5.4186
 Median :2017-04-05   AVENIDA DEL MEDITERRANEO, S/N:   561   41500  :  2940   L-D: 07:00-23:00:132168   Median :40.22   Median : -3.3903
 Mean   :2017-04-05   AVENIDA DE GRANADA, S,N      :   510   29680  :  2805   L-D: 07:00-22:00: 43251   Mean   :39.68   Mean   : -3.2364
 3rd Qu.:2017-04-18   AUTOPISTA AP-8 KM. 22        :   480   30800  :  2777   L-D: 06:00-23:00: 39552   3rd Qu.:41.76   3rd Qu.: -0.5182
 Max.   :2017-05-02   AVENIDA CONSTITUCION, S/N    :   480   29600  :  2652   L-D: 06:00-00:00: 35994   Max.   :43.69   Max.   :  4.2795
                      (Other)                      :1311442   (Other):1297658   (Other)         :272439   NA's   :102     NA's   :102
     localidad            margen          municipio          rotulo                provincia      media_precio                tipo_gasol
 MADRID   : 25072   D       :637781   Madrid   : 25072   REPSOL  :396798   BARCELONA          :104076   Min.   :0.499   gasoleoA             :448932
 BARCELONA: 12459   I       :365702   Murcia   : 13379   CEPSA   :191049   MADRID             : 90848   1st Qu.:1.127   gasolina95Proteccion :432352
 SEVILLA  :  8132   N       :311981   Barcelona: 12459   GALP    : 71398   VALENCIA / VALÈNCIA: 75324   Median :1.201   gasolina98           :301230
 VALENCIA :  7765           :     0   Valencia :  8581   SHELL   : 46779   ALICANTE           : 52373   Mean   :1.179   gasoleoB             :102187
 MALAGA   :  7131   -0.065389:     0   Sevilla  :  8234   PETRONOR: 28981   MURCIA             : 52136   3rd Qu.:1.293   gasesLicuadosPetroleo: 23980
 ZARAGOZA :  6505   -0.11275 :     0   Cartagena:  8165   BP      : 28854   SEVILLA            : 47000   Max.   :1.699   biodiesel            :  3301
 (Other)  :1248400  (Other)  :     0   (Other)  :1239574   (Other) :551605   (Other)            :893707                   (Other)              :  3482
```

Figura 9: Descriptivos más comunes de los datos de los precios de los carburantes

En la Figura 8, vemos que en las 3 primeras filas coinciden todos los valores de las variables menos la “fecha”, “precio” y “tipo_gasol”, y, por tanto, en la Figura 9 encontramos 1139 casos en los que existe una gasolinera en la dirección “Avenida Andalucía S/N”, y lo mismo ocurre con las demás variables. Por lo que vamos a proceder a crear otro conjunto de datos, el cual llamaremos “gasolineras”, en el que eliminaremos esas variables dinámicas, como son, fecha, y precio, para así poder referirnos a las gasolineras (hablamos de una gasolinera cuando solo se encuentra en una dirección, código postal, latitud, y longitud). Esto es debido a la forma en la que extrajimos los datos ya que era la forma más sencilla de hacerlo.

Por tanto, cuando nos refiramos a los precios de los carburantes, usaremos el fichero inicial, el llamado “precios_gasol_def”, en el cual se encuentran las variables dinámicas (Figura 8). Mientras que cuando nos refiramos a las gasolineras usaremos el conjunto de datos “gasolineras”, el cual consta de 11 variables y 26551 observaciones.

Un subconjunto de este bastidor se puede ver en la Figura 10.

	direccion	cp	horario	latitud	longitud	localidad	margen	municipio	rotulo	provincia	tipo_gasol
1	CL MANISITU, 9	01240	L-D: 24H	42.84603	-2.509361	ALEGRIA-DULANTZI	D	Alegría-Dulantzi	REPSOL	ÁLAVA	gasoleoA
2	CL MANISITU, 9	01240	L-D: 24H	42.84603	-2.509361	ALEGRIA-DULANTZI	D	Alegría-Dulantzi	REPSOL	ÁLAVA	gasoleoB
3	POLIGONO ZANKUETA, 0	01468	L-D: 24H	43.04433	-2.989111	LARRINBE	D	Amurrio	ESTACIONES GB	ÁLAVA	gasoleoA
4	POLIGONO ZANKUETA, 0	01468	L-D: 24H	43.04433	-2.989111	LARRINBE	D	Amurrio	ESTACIONES GB	ÁLAVA	gasolina95Proteccion
5	POLIGONO ZANKUETA, 0	01468	L-D: 24H	43.04433	-2.989111	LARRINBE	D	Amurrio	ESTACIONES GB	ÁLAVA	gasolina98
7	CARRETERA A-624 KM. 37,8	01450	L-D: 06:00-22:00	43.03189	-2.967611	LEZAMA	D	Amurrio	CEPSA	ÁLAVA	gasoleoA

Figura 10: Subconjunto de los datos de las gasolineras, “gasolineras”

4 Objetivos

En este apartado, vamos a establecer a priori los objetivos fundamentales que se quieren conseguir con este estudio. Entre ellos, vamos a distinguir entre objetivo general y secundario.

4.1 Objetivo general

El objetivo general de este estudio es el conocimiento, estudio y análisis del comportamiento de los precios de los carburantes de España. Buscaremos comprender la relación del precio de los distintos establecimientos con respecto a las distintas características físicas de cada gasolinera, como son, la dirección, la latitud, la longitud, la provincia, el municipio, para saber la ubicación de esa gasolinera, el rótulo, el tipo de gasolina que se vende, entre otras, cuyo fin es entender cómo las características mencionadas anteriormente afectan a la gasolinera a la hora de establecer un precio. Conocer estas relaciones será fundamental para llegar a comprender el comportamiento de las distintas gasolineras teniendo en cuenta la competencia, ya que un establecimiento marca sus tarifas atendiendo no sólo a sus características individuales, sino también a las del resto de empresas.

4.2 Objetivos secundarios

Con objeto de perseguir el objetivo anterior, han surgido los siguientes objetivos secundarios, los cuales son:

- Conocer a fondo la información de partida a partir de un profundo análisis descriptivo.
- Establecer una metodología de las técnicas utilizadas durante todo el estudio.
- Agrupar las variables en grupos homogéneos y distintos entre sí.
- Limpiar los datos de algunas de las variables.
- Análisis de datos faltantes y atípicos.
- Interpretar y comprender las distintas variables en uso. Para ello, nos apoyaremos en técnicas multivariantes como son, el análisis clúster.
- Dividir la base de datos en dos partes (Entrenamiento, y Test) con la finalidad de conseguir modelos más óptimos en cuanto a su validez predictiva.
- Obtener distintos modelos de predicción de los precios de las gasolineras de España usando varias técnicas de Machine Learning, como son, Regresión Lineal, Redes Neuronales, Random Forest, y Suport Vector Machine.
- Establecer una comparativa que permita señalar el mejor modelo predictivo obtenido.
- Crear 8 variables de la competencia referidas al precio medio y mínimo de aquellas gasolineras que se encuentran a una determinada distancia. Las distancias serán 5, 10, 20, y 50 km.
- Estudiar del comportamiento de una gasolinera, en comparación con su competencia, usando teoría de juegos, más concretamente, el Juego de Bertrand.
- Determinar la zona en la que aplicar el punto anterior. En este caso, será La Gomera.
- Encontrar una función de demanda de acuerdo con las distintas gasolineras.
- Fijar de alguno de los parámetros de la función de demanda, atendiendo a la competencia y en distintas situaciones.
- Estudiar el precio de los carburantes de La Gomera.

5 Metodología

En este apartado, vamos a explicar de forma breve la metodología en la que nos hemos basado, la cual se llama Metodología SEMMA, y las técnicas estadísticas que hemos utilizado para poder llevar a cabo los objetivos expuestos anteriormente, además del software que hemos empleado.

5.1 Metodología SEMMA

Para alcanzar los objetivos establecidos, se ha intentado seguir el patrón de trabajo establecido por la Metodología SEMMA (Sample, Explore, Modify, Model, Asses, la cual se puede observar en la Figura 1.

El Sample (Muestrear) se refiere a realizar un muestreo de los datos, en el que se debe seleccionar un conjunto de datos lo más representativo de la población y lo suficientemente grande. El Explore (Explorar) se refiere a descubrir anomalías, y relaciones en los datos. El Modify (Modificar) se basa en seleccionar, crear, y transformar variables. El Model (Modelizar) se refiere a la creación de distintos modelos usando distintas técnicas. El Asees (Evaluar) se refiere a la evaluación de los resultados de los distintos modelos realizados en la fase anterior.

Aunque el esquema de la Figura 1 [19] es el que hay que seguir, no ha de seguirse exhaustivamente, ni en orden ni en contenido, ya que:

- No siempre intervienen todas las fases del proceso (por ejemplo, muestrear en el presente estudio no se ha llevado a cabo).
- A menudo el orden no es exacto (se puede explorar antes de muestrear; se puede explorar-modificar-explorar-modificar, etc.)
- El proceso se suele repetir muchas veces, pasando de unas fases a otras sin respetar el orden del flujo.

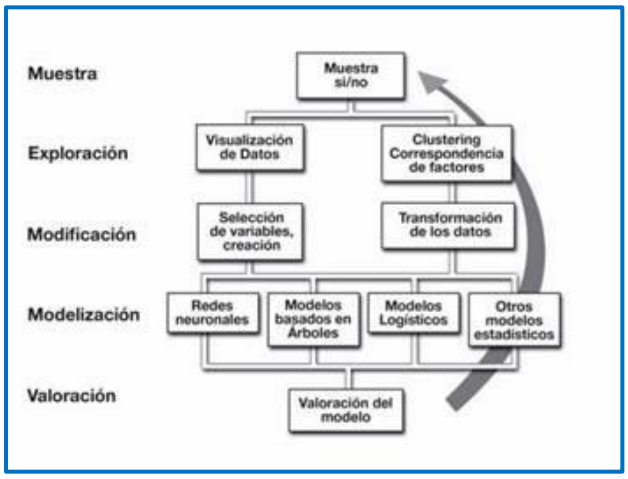


Figura 1: Metodología SEMMA.
FUENTE:http://actividadsenadisenocubosdedatos.blogspot.com.es/2015/11/metodologias-para-mineria-de-datos.html

5.2 Distancia de semiverseno

La distancia de semiverseno [30] sirve para calcular la distancia de círculo máximo entre dos puntos de un globo sabiendo su longitud y latitud. Es decir,

para cualquier par de puntos sobre una esfera, la distancia de semiverseno se calcula como:

$$\text{haversin}(d/R) = haversin(\varphi_1 - \varphi_2) + cos(\varphi_1)cos(\varphi_2)\, haversin(\nabla\gamma)$$

donde:

- $haversin$ es la función haversine, $haversine(\theta) = sen^2(\theta/2) = (1 - cos(\theta))/2$
- d es la distancia entre dos puntos
- R es el radio de la esfera
- φ_1 es la latitud del punto 1
- φ_2 es la latitud del punto 2
- $\nabla\gamma$ es la diferencia de longitudes

5.3 Test de Grubbs

Dada una muestra aleatoria de tamaño n procedente de una población univariante, $(x_1, x_2, \ldots, x_n)$. Con este test podemos sospechar si una de las observaciones es un dato atípico.

Por tanto, el test de Grubbs [16] sirve para detectar datos atípicos en una muestra, en el que se exige que la muestra proceda de una población normal.
El contraste de este test es el siguiente:

$$H_0: No\ hay\ atípicos\ en\ la\ muestra$$
$$H_1: Hay\ al\ menos\ un\ atípico\ en\ la\ muestra$$

Para ello, se usará el estadístico $G = \frac{max\,|x_i - \bar{x}|}{\sigma}$, siendo:

$\bar{x}$ la media
σ la desviación típica muestrales

La región crítica de este contraste se puede obtener aproximadamente tomando como referencia la distribución t_{n-2} de Student con $(n - 2)$ grados de libertad. Así, definiendo k tal que,

$$P\,(t_{n-2} > k) = {}^{\alpha}/_{2n}, \text{ siendo } \alpha\ el\ nivel\ de\ significación$$

se aceptará la hipótesis alternativa H_1 de existencia de dato atípico si G excede de cierto valor crítico:

$$G > \frac{n - 1}{\sqrt{n}} \sqrt{\frac{k^2}{n - 2 + k^2}}$$

5.4 Análisis Clúster

El análisis clúster utiliza [3] la información de una serie de variables para cada sujeto u objeto y, conforme a estas variables, mide la similitud entre ellos. Una vez medida la similitud se organizan en grupos homogéneos internamente y diferentes entre sí. La

"nueva dimensión" lograda con el clúster se aprovecha después para facilitar la aproximación "segmentada" de un determinado análisis.

A la hora de aplicar esta técnica, conviene tener en cuenta que tiene propiedades inferenciales, y que, como tal, los resultados obtenidos para una muestra sólo sirven para su diseño.
El análisis clúster trata de resolver el siguiente problema: dado un conjunto de elementos, caracterizados por la información de variables, n variables, X_j, se busca poder clasificarlos de manera que los individuos pertenecientes a un grupo o clúster sean tan similares entre sí como sea posible, atendiendo también a que los distintos grupos deben poder diferenciarse al máximo, en la medida de lo posible [4].

Este proceso consta de tres fases, que se pueden resumir en:

- Establecer un criterio de similitud para poder determinar una matriz de parecidos que nos permita relacionar la semejanza de los individuos entre sí.
- Aplicación de un algoritmo de clasificación con el que determinar la estructura de agrupación de los individuos.

Una vez seleccionadas las variables a considerar, cada uno de los individuos sujetos al análisis vendrá representado por los valores que tomen estas variables en cada uno de ellos. Este es el punto de partida de la clasificación. Para clasificar adecuadamente los individuos se debe determinar lo similares o disimilares (divergentes) que son entre sí, en función de lo diferentes que resulten ser sus representaciones en el espacio de las variables [5].

Para medir esta similitud, existen distintos índices, todos ellos con propiedades y utilidades diferentes, cuyo uso habrá que considerar en cada caso. La mayor parte de estos índices serán o bien, indicadores basados en la distancia (considerando a los individuos como vectores en el espacio de las variables); indicadores basados en coeficientes de correlación; o bien basados en tablas de datos de posesión o no de una serie de atributos.

En nuestro caso, nos basaremos en el concepto de distancia a la hora de medir la similitud entre los distintos objetos. Daremos el nombre de distancia entre dos individuos i y j a la medida $d(i,j)$, que indicará el grado de semejanza o desemejanza entre los objetos mencionados, en relación a un cierto número de características. Este valor, siempre debe cumplir:

1) $d(i,j) > 0$
2) $d(i,j) = 0$
3) $d(i,j) = d(j,i)$

En general, nos referiremos a la distancia euclídea como unidad de medida (aunque existen otras, como la distancia de Mahalanobis, o la distancia de Minkoswski, entre otras), la cual, además de las premisas mencionadas, verifica

4) $d(i,j) < d(i,t) + d(t,j)$

5) $d(i,j) > 0 \ \forall \ i,j$

y cuya fórmula es: $d(X_i, X_{i'}) = \sqrt{\sum_{j=1}^{p}(X_{ij} - X_{i'j})^2}$

En el presente estudio, se aplicará un método de clúster no jerárquico, algoritmo de las k-Medias, basado en la media de las variables en uso y mediante pruebas sucesivas contrasta el efecto que sobre la varianza residual tiene la asignación de cada una de las observaciones de cada grupo. La de mínima varianza es la que determina la asignación. Este paso se repite hasta que todas las observaciones hayan sido asignadas. A continuación, se determina el centroide de todos los grupos y se repite el proceso de asignación nuevamente desde la primera a la última observación. En el que se formarán grupos homogéneos sin establecer relaciones entre ellos [6].

5.5 Regresión lineal

Los modelos de regresión lineal [7] son modelos matemáticos usados para aproximar la relación de dependencia entre una variable dependiente Y, y las variables independientes X_iy un término aleatorio ε. Este modelo puede ser expresado como

$$Y_t = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon$$

donde:

- Y_t es la variable dependiente
- X_i son las variables explicativas o independientes
- β_i son los parámetros respectivos a cada variable independiente que miden la influencia que las variables explicativas tienen, siendo β_0 el término constante.
- ε es una variable aleatoria, que normalmente se supone que es una $N(0, \sigma)$

El objetivo de la regresión es escoger unos valores determinados para los parámetros β_i, de modo que la ecuación anterior quede completamente especificada, de modo que pueda quedar determinada la relación existente entre la variable dependiente y las explicativas.

Para poder hacer esta determinación, deben cumplirse ciertos supuestos:

- Relación lineal entre variables.
- Independencia entre los errores obtenidos en la medición de las variables explicativas.
- Varianza constante en los errores (homocedasticidad).
- Esperanza igual a 0 en los errores.
- La suma de todos los errores debe ser idéntica al error total.

A la hora de comparar los distintos modelos, se pueden considerar distintas medidas de ajuste, las cuales evalúan en qué medida el modelo utilizado explica las variaciones que se producen en la variable dependiente.

Por tanto, las medidas de ajuste que se utilizan son, principalmente,

- $R^2 = 1 - \frac{SSE}{SST}$
- $R^2\ ajustado = 1 - \frac{(n-1)\,(1-R^2)}{(n-p)}$
- $BIC = n\ ln(\frac{SSE}{n}) + 2(p+2)q - 2q^2$, donde $q = \frac{n\sigma^2}{SSE}$
- $AIC = n\ ln(\frac{SSE}{n}) + 2p$
- $SBC = n\ ln(\frac{SSE}{n}) + p\ ln(p)$
- $ASE = \frac{SSE}{n}$
- $RMSE = \sqrt{ASE}$

donde: $SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
$SST = \sum_{i=1}^{n}(y_i - \bar{y}_i)^2$

siendo,

- y_i son los datos originales
- $\hat{y}_i$ son las predicciones
- $\bar{y}_i$ son las predicciones corregidas
- n es el número de observaciones

Excepto R^2 y $R^2\ ajustado$, todas estas medidas se consideran mejores cuanto más pequeñas sean. En nuestro caso, nos fijaremos en el RMSE normalizado o error de raíz cuadrática media o error de predicción.
Las principales ventajas [8] de la regresión lineal son:

- El análisis de regresión es una herramienta muy flexible en cuanto a la naturaleza de las variables explicativas, pues éstas pueden ser numéricas y categóricas.
- Permite hacer una predicción del comportamiento de alguna variable en un determinado punto o momento.

5.6 Redes neuronales

Las Redes Neuronales [9] constituyen una herramienta muy potente de análisis, modelización y predicción. Se rigen por la filosofía general de obtener modelos coherentes con la realidad observada, de tal modo que sean los datos los que determinen el comportamiento de la red, ya sea a través de la determinación de sus estructuras o de sus parámetros internos. Esta técnica forma parte de los métodos no paramétricos de análisis de datos.

Las redes neuronales están basadas en el modo que funcionan las neuronas en el cerebro. La operación general de la misma es transmitir químicos dentro del fluido del cerebro para aumentar o disminuir el potencial eléctrico dentro del cuerpo de la neurona. Si el potencial de la neurona alcanza algún determinado umbral, la neurona se activa y un pulso de duración fija se envía al axón. El axón se divide en conexiones a muchas otras neuronas, conectando a estas neuronas en una sinapsis (Figura 2).

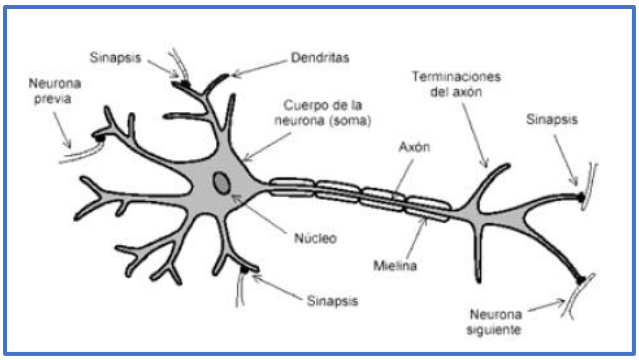


Figura 2: Estructura de una neurona

Se propuso un modelo matemático simplificado del funcionamiento de una neurona con las siguientes características (Figura 3):

- Un conjunto de pesos w_i que corresponden con la sinapsis.
- Un sumador Σ que suma las señales entrantes (equivalente a la membrana de la célula que recoge la carga eléctrica).
- Una función de activación g que decide si la membrana se activa o no para las entradas actuales.

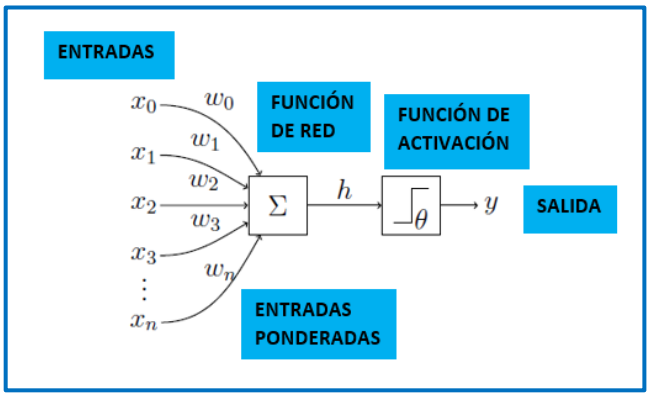


Figura 3: Esquema del funcionamiento de una red

donde:

h a la suma de las entradas multiplicadas por los pesos. Es decir,

$$h = \sum_{i=1}^{n} w_i \, x_i$$

Si $h > \theta$ (valor fijado), la neurona se activará. Por lo que sólo puede activarse o no hacerlo, por lo que no puede aprender. Para ello, necesitamos poner neuronas juntas formando una red neuronal. Por ello, surgió el Perceptrón Multicapa, que consiste en múltiples capas de neuronas (Figura 4).

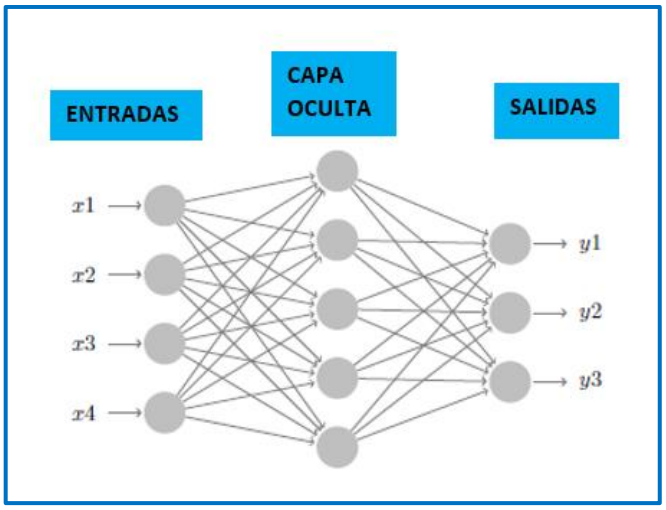


Figura 4: Perceptrón multicapa

El teorema de aproximación universal establece que un perceptrón multicapa con una capa oculta puede aproximar funciones continuas [11].

Además, las Redes Neuronales presentan grandes ventajas [10] que nos llevan a elegir este método de predicción en lugar de otros. Algunas de ellos son:

- Aprendizaje adaptativo: no es necesario elaborar modelos a priori ni especificar funciones de distribución de probabilidad.
- Autoorganización: que provoca la generalización.
- Tolerancia a fallos.
- Operación en tiempo real.
- Fácil inserción dentro de la tecnología existente.

Por otro lado, esta técnica de predicción presenta la desventaja de la pérdida de interpretabilidad de los resultados, donde solo se puede evaluar la importancia de cada una de las variables explicativas del modelo. Creándose un ranking de las variables según su frecuencia utilizada en el algoritmo.

5.7 Random Forest

Random Forest [12] es un algoritmo de combinación de árboles predictores, tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos basado en árboles de decisión y de clasificación. En este caso, lo vamos a utilizar como árbol de decisión [13] ya que nuestra variable predictora es de carácter continuo.

A continuación, se presenta en la Figura 5 el esquema de este algoritmo.

Dados los datos de tamaño N.

1. Repetir m veces i), ii), iii):

   i) Seleccionar n<N observaciones con reemplazamiento de los datos originales.

   ii) Aplicar un árbol de la siguiente manera:

   - En cada nodo, seleccionar p variables de las k originales y de las p elegidas, escoger la mejor variable para la partición del nodo.

   iii) Obtener predicciones para todas las observaciones originales N.

2. Promediar las m predicciones obtenidas en el apartado 1).

Figura 5: Pasos del algoritmo de Random Forest

En general, los parámetros a tener en cuenta en este algoritmo son:

- El tamaño de las muestras, n, y si se va a utilizar bootstrap (con reemplazo) o sin reemplazamiento.
- El número de iteraciones a promediar, m.
- Características del árbol: número de hojas, profundidad, el número de divisiones máximas en cada nodo, el p-valor para las divisiones en cada nodo, y el número de observaciones mínimas en cada rama-nodo.
- Número de variables a muestrear en cada nodo, p.

Este algoritmo incorpora dos fuentes de variabilidad (remuestreo de observaciones y de variables) para mejorar la capacidad de generalización, y reducir el sobreajuste, conservando en cualquier caso la facultad de ajustar bien las relaciones particulares de los datos (interacciones, no linealidad, cortes, problemas de extrapolación, etc.)

Las principales ventajas de esta técnica son:

- Aumenta la capacidad predictiva y disminuye la varianza.
- Disminuye la sensibilidad frente a cambios en los datos, aumenta la estabilidad y la robustez.
- Aumenta la suavidad (función menos escalonada), lo que a veces redunda en menor error promedio de predicción.

Por otro lado, esta técnica de predicción presenta la desventaja de la pérdida de interpretabilidad de los resultados, donde solo se puede evaluar la importancia de cada una de las variables explicativas del modelo. Creándose un ranking de las variables según su frecuencia utilizada en el algoritmo.

5.8 Suport vector machine

Los Support Vector Machine (SVM) [11] son uno de los algoritmos más populares en machine learning. Se han empleado en multitud de aplicaciones desde entonces, debido

principalmente a que consigue un menor error en conjuntos de datos con tamaño razonable. Los SVM no funcionan bien en conjuntos de datos muy grande, ya que los cálculos no escalan bien con el número de datos, y se vuelve computacionalmente muy costoso.

Los SVM se basan en la función de pérdida que ignora los errores, que están situados dentro de la cierta distancia del verdadero valor. Este tipo de función a menudo se llama - épsilon intensivo - función de pérdida.

En la Figura 6 se muestra un ejemplo de función de regresión lineal unidimensional con épsilon intensivo. Las variables miden el costo de los errores en los puntos de entrenamiento. Estos son cero para todos los puntos que están dentro de la banda [14].

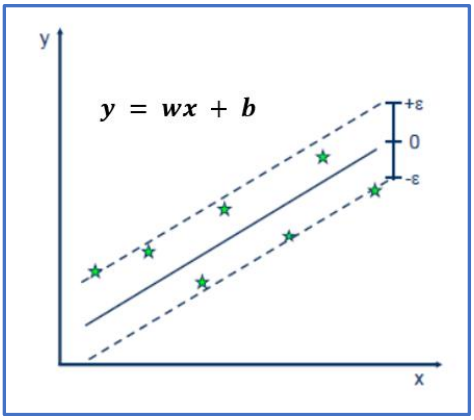


Figura 6: Ejemplo de SVM

$$min\ \frac{1}{2}\ ||w||^2$$
$$s.a\ y_i\ -\ wx_i\ -\ b \leq \varepsilon$$

donde:

x es el vector de entradas
w es un vector perpendicular al hiperplano clasificador
b es una constante
y_i es el valor de la salida i-ésima

Otro concepto fundamental en SVM es el de kernel, que consiste en transformar los datos en un espacio de características dimensionales más altas para posibilitar la separación lineal (Figura 7).

Un kernel K es una función que de define como $K(x, y) = \varphi(x) \cdot \varphi(y)$, donde x, y son vectores de entradas, φ es una función a un espacio de dimensión superior al de entrada.

Alguno de los kernels más habituales son los siguientes:

- Polinomial de grado d,

$$K(x, y) = (x * y)^d$$

- Radial con parámetro σ,

$$K(x, y) = exp\ (-\frac{||x - y||^2}{2\sigma^2})$$

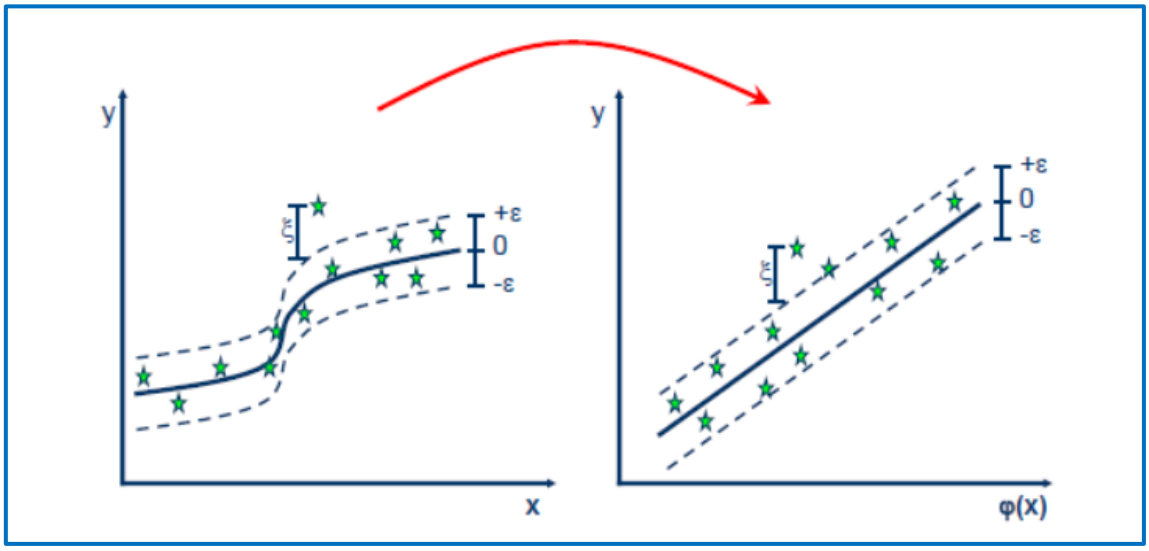


Figura 7: Kernel

Por otro lado, esta técnica de predicción presenta la desventaja de la pérdida de interpretabilidad de los resultados.

5.9 Validación cruzada repetida

La validación cruzada repetida [15] es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar la precisión de un modelo que se llevará a cabo a la práctica.

Esta técnica consiste en que los datos de muestra se dividen en K subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto $(K - 1)$ como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Finalmente se realiza la media aritmética de los K valores de errores obtenidos de cada iteración para obtener un único resultado (Figura 8).

Este método es muy preciso puesto que evaluamos a partir de K combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja, y es que, es lento desde el punto de vista computacional.

En la práctica, la elección del número de iteraciones depende de la medida del conjunto de datos.

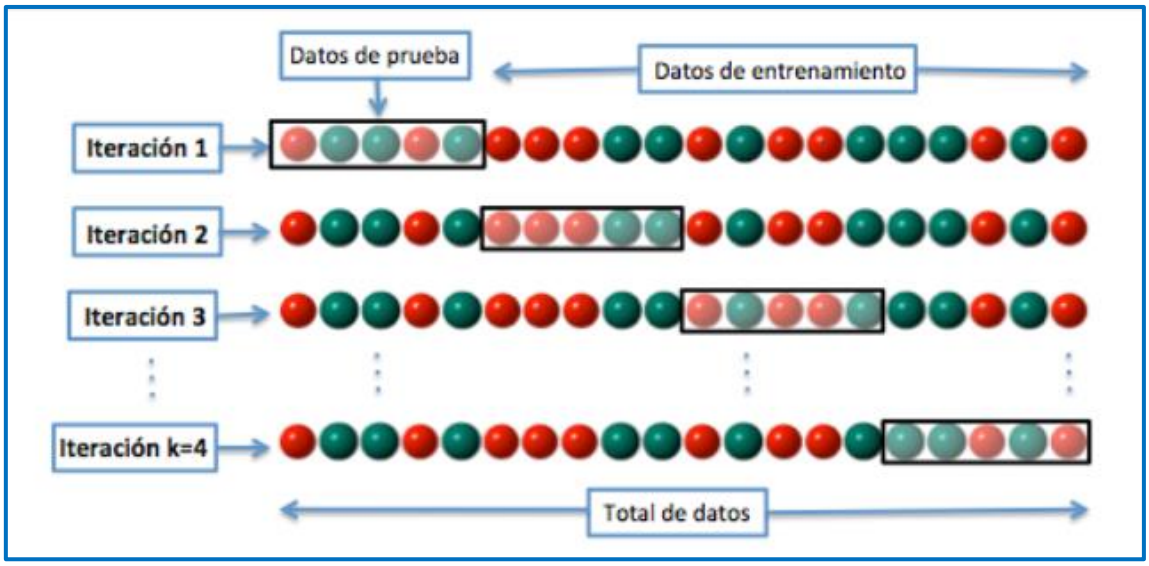


Figura 8: Procedimiento de la validación cruzada repetida

5.10 Comparación de modelos

Una vez aplicadas todas las técnicas mencionadas anteriormente, nuestra finalidad es escoger cuál de esas técnicas se ajusta mejor a nuestros datos, es decir, elegiremos el modelo óptimo.

Para ello, nos vamos a basar en el ASE o error cuadrático medio en los datos de validación cruzada repetida. Como bien comentamos antes, escogeremos aquel modelo que tenga un menor error, además de contar con la opinión personal del investigador, a la hora de elegir entre estructuras más complejas.

5.11 Teoría de juegos: Juego de Bertrand

El Juego de Bertrand busca la mejor estrategia, con mejor estrategia nos referimos a precio óptimo, para cada jugador teniendo en cuenta la elección de la competencia, o lo que es lo mismo, de los otros jugadores [20].

En nuestro estudio, los jugadores serán cada una de las gasolineras que se encuentren en una zona de España, en este caso, La Gomera. Es decir, los jugadores serán cada una de las gasolineras que existan en La Gomera.

Esta estrategia se basa en el Equilibrio de Nash (EN), que consiste en que todos los jugadores escogen una estrategia que maximiza sus ganancias dadas las estrategias de los otros. Es decir, si dando por hecho lo que hicieron los demás, un jugador puede mejorar su situación cambiando su comportamiento, no estamos en un equilibrio de Nash [21].

Además, añadir que este modelo se caracteriza porque utiliza una función de demanda, la cual se fija para determinar el grado de sustitución que tiene la competencia en mi producto, y donde la elección estratégica se basa en el precio [18].

A continuación, vamos a mostrar un ejemplo práctico de este juego [17] teniendo 2 empresas (duopolio), que es el caso más fácil de explicar.

Asumiendo que no existen costes fijos de producción y que los costes marginales son constantes e iguales a c, la función de demanda de la empresa 1 si las empresas 1 y 2 eligen los precios p_1 y p_2 respectivamente sería:

$$q_1(p_1, p_2) = a - p_1 + b * p_2$$

donde:

- a es el término constante, siendo $a > c$
- b refleja hasta qué punto el producto de la empresa 1 es un sustituto de la empresa 2
- p_1 y p_2 son los precios de la empresa 1 y de la 2, respectivamente
- $q_1(p_1, p_2)$ es la función de demanda de la empresa 1

Para la empresa 2, esta función sería de forma equivalente, pero con los índices correspondientes.

Vamos a asumir que fabricar una unidad cuesta c unidades tendremos el siguiente juego:

- Número de jugadores: 2
- Estrategias: $p_1 \in (0, \infty)$, $y\ p_2 \in (0, \infty)$
- Función de beneficios de cada una de las empresas:

$$\pi_1(p_1, p_2) = p_1 * q_1(p_1, p_2) - c*q_1(p_1, p_2) = (p_1 - c) * q_1(p_1, p_2)$$
$$\pi_2(p_1, p_2) = p_2 * q_2(p_1, p_2) - c*q_2(p_1, p_2) = (p_2 - c) * q_2(p_1, p_2)$$

Una vez definido el juego, se calcula la función de mejor respuesta de ambos jugadores. Para ello se busca el máximo de la función de pagos de un jugador fijando el otro jugador.

Función de mejor respuesta de J_1:

$$\frac{d\,\pi_1(p_1,p_2)}{dp_1} = \frac{d\,((p_1 - c) * q_1(p_1,p_2))}{dp_1} = (a - p_1 + b * p_2)(p_1 - c) = 0$$

$$\Rightarrow \quad p_1 = \frac{a + c + bp_2}{2}$$

En este caso se puede observar que, si la competencia sube los precios, la empresa a analizar también lo subirá y viceversa. La intensidad depende del valor de b, si éste baja, es una competencia lejana y, por tanto, la empresa a analizar le afecta poco. Mientras que si es competencia directa (valor de b alto) le afectará mucho más.

Función de mejor respuesta de J_2:

$$\frac{d\,\pi_2(p_1,p_2)}{dp_2} = \frac{d\,((p_2 - c) * q_2(p_1,p_2))}{dp_2} = (a - p_2 + b * p_1)(p_2 - c) = 0$$

$$\Rightarrow \quad p_2 = \frac{a + c + bp_1}{2}$$

Como se puede observar, el resultado es totalmente simétrico al caso anterior.

Por último, vamos a calcular el EN que será el punto de corte de las 2 funciones de mejor respuesta. Es decir,

$$\begin{cases} p_1 = \dfrac{a + c + bp_2}{2} \\ p_2 = \dfrac{a + c + bp_1}{2} \end{cases}$$

Resolviendo este sistema,

$$p_1 = \frac{a + c}{2 - b}$$

$$p_2 = \frac{a + c}{2 - b}$$

Por tanto, el Equilibrio de Nash sería $(p_1, p_2) = (\frac{a+c}{2-b}, \frac{a+c}{2-b})$.

5.12 Software empleado

Todos los métodos y procesos que se realizan en el siguiente trabajo se han realizado con el software libre RStudio.

El principal motivo por el que se ha utilizado este software es porque es un software libre y por lo tanto puede ser usado por toda la comunidad son coste alguno, y por tener un mayor conocimiento en este software a la hora de elaborar código.

6 Análisis descriptivo de las variables

En este apartado, vamos a realizar una breve descripción de cada una de las variables de nuestra base de datos para tener un mayor conocimiento de la información de la que partimos, haciendo una distinción entre las variables propiamente descriptivas de una gasolinera, como son, la dirección, la latitud, la longitud, el código postal, su rótulo, entre

otras; y variables de la competencia, referidas al precio de aquellas gasolineras que se encuentran a una determinada distancia, las cuales han sido creadas y mostraremos a continuación.

Para las variables de naturaleza cuantitativa se usarán gráficos de barras como el histograma o descriptivos como la media, mientras que para las variables de naturaleza categórica se estudiará la evolución de la frecuencia de cada una de sus categorías.

Además, hay que tener en cuenta que en nuestra base de datos contamos con variables estáticas, que son fijas de cada gasolinera como por ejemplo su ubicación o el horario, para las cuales usaremos el conjunto de datos "precios_gasol_def", y otras dinámicas, como el precio de la gasolina, para las que usaremos el bastidor "gasolineras", como bien hemos comentado anteriormente.

6.1 Variables descriptivas de una gasolinera

Como hemos mencionado antes, vamos a explicar las variables propiamente descriptivas de una gasolinera, entre las que vamos a distinguir entre variables cuantitativas y cualitativas (en el Anexo A podemos encontrar los gráficos correspondientes).

6.1.1 Variables cuantitativas

Se consideran variables cuantitativas todas aquellas que toman como argumento cantidades numéricas ya sean discretas o continuas.

En este caso, son las siguientes:

- Latitud (estática) (Anexo A, Figura 1)

  Podemos observar que la mayoría de las gasolineras tienen una latitud entre 35 y 42, aproximadamente, ya que son las que corresponden a la península. Mientras que las menos frecuentes, abarcan desde 1 hasta 28, aproximadamente, corresponden a la de las islas Canarias.

- Longitud (estática) (Anexo A, Figura 2)

  En este caso, ocurre lo mismo que en la latitud. Es decir, la mayoría de las gasolineras de nuestro estudio corresponden a la península, mientras que el resto de ellas pertenecen a las islas Canarias.

- Precio (dinámica) (Anexo A, Figura 3)

  Esta variable, la vamos a observar en función del tipo de gasolina ya que, dependiendo del tipo de gasolina, el precio varía.

Podemos observar que, a simple vista, la variable precio tiene valores atípicos, los cuales comprobaremos más adelante. Además, del tipo de gasolina bio (Biodiésel, y Bioetanol), el precio más barato es el de Biodiésel; entre los gases (Gases Licuados Petróleo, Gas Natural Comprimido, y Gas Natural Licuado), el precio más barato es el de Gases Licuados Petróleo; en el caso de gasoil (Gasóleo A, y Gasóleo B), sería Gasóleo B; y en el caso de gasolina (Gasolina 98, y Gasolina 95 Protección), sería Gasolina 95 Protección.

6.1.2 Variables cualitativas

Se consideran variables cualitativas a todas aquellas variables que toman como argumento una clasificación o modalidad. Cada modalidad que se presenta se denomina atributo o categoría y la medición consiste en una clasificación de dichos atributos.

En este caso, son las siguientes:

- Código postal (estática) (Anexo A, Figura 4 y 5)

  En este caso, como tenemos un gran número de niveles, exactamente 4323, vamos a representar en el gráfico los 9 códigos postales más frecuentes, los cuales son el 04700, el 30500, 41500, 29680, 30800, 29200, 29600, 35500, 46500, que pertenecen a Almería, Murcia, Sevilla, Málaga, Marbella, Las Palmas, y Valencia.

- Horario (estática) (Anexo A, Figura 6 y 7)

  En este caso, como tenemos un gran número de niveles, exactamente 557, vamos a representar en el gráfico los 20 horarios de las gasolineras de España más frecuentes. De esos horarios, la mayoría de las gasolineras abren de lunes a domingo las 24 h, o de 6 a 22 h.
  Esta variable más adelante la categorizaremos en 2 niveles, “abierto las 24h” y “no abre las 24h”, ya que podemos aglutinar la misma información en categorías iguales.

- Localidad (estática) (Anexo A, Figura 8, y 9)

  En este caso, como tenemos un gran número de niveles, exactamente 557, vamos a representar en el gráfico las 39 localidades más frecuentes. De estas, la mayoría de gasolineras se encuentran en Madrid y Barcelona, ya que son las ciudades más grandes de España.

- Margen (estática) (Anexo A, Figura 10)

  La mayoría de las gasolineras están situadas a la derecha de una carretera.

- Municipio (estática) (Anexo A, Figura 11 y 12)

En este caso, como tenemos un gran número de niveles, exactamente 3146, vamos a representar en el gráfico los 47 municipios más frecuentes. De estos, la mayoría de gasolineras se encuentran en Madrid, Barcelona, y Murcia.

- Rótulo (estática) (Anexo A, Figura 13 y 14)

  En este caso, como tenemos un gran número de niveles, exactamente 3543, vamos a representar en el gráfico los 20 nombres de las gasolineras de España más frecuentes. De estos, la mayoría de las gasolineras son REPSOL y CAMPSA
  Esta variable más adelante la categorizaremos en 9 niveles, REPSOL, CEPSA, GALP, SHELL, BP, PETRONOR, y CAMPSA, (las más frecuentes) gasolineras de los supermercados como Carrefour o Eroski en "Supermercados", y para los demás vamos a categorizarles en la categoría de "Otros".

- Provincia (estática) (Anexo A, Figura 15)

  La mayoría de las gasolineras se encuentran en Madrid, Barcelona, Murcia y Valencia.

- Tipo gasolina (estática) (Anexo A, Figura 16)

  Los tipos de gasolina más comunes en España son gasóleo A, gasolina 95 protección, y gasolina 98, los cuales son los que usan la mayoría de los automóviles.

6.2 Variables de la competencia

Las variables de la competencia están referidas al precio medio y precio mínimo de aquellas gasolineras que se encuentran a una determinada distancia. En este caso, las distancias son 5, 10, 20 y 50 km. Es decir, vamos a obtener el precio medio de aquellas gasolineras que se encuentran a 5 km, la cual llamaremos "precio_5km" para poder observar la competencia. "Precio_10km" son los precios medios de las gasolineras que se encuentran a 10 km, y así con las demás. Por lo que tendremos 4 variables referidas al precio medio a 5, 10, 20, y 50 km; y otras 4, referidas al precio mínimo a las mismas distancias, 5, 10, 20, y 50 km. Tanto para los precios medios como para los precios mínimos, los llamaremos de igual forma. Estas 8 variables las hemos construido a través de la distancia de semiverseno.

Estas variables son de naturaleza cuantitativa.

Esto lo hemos realizado por tipo de gasolina, aunque prevemos que más o menos nos dará un parecido resultado. De todos los tipos de gasolina (Gasóleo A, Gasóleo B, Gasolina 95 Protección, Gasolina 98, Biodiésel, Bioetanol, Gases Licuados Petróleo, Gas Natural Comprimido, Gas Natural Licuado) lo hemos realizado solo para un tipo de gasolina, en este caso, hemos escogido Gasolina 98, y para un tipo de gasoil, Gasóleo A.

En primer lugar, vamos a mostrar las variables creadas, para después realizar el análisis descriptivo.

En la Figura 17 hasta la Figura 24, se puede observar una submuestra para cada una de las 4 variables creadas de los precios medios de las gasolineras a 5, 10, 20, y 50 km, “precio_5km”, “precio_10km”, “precio_20km”, y “precio_50km”, tanto para el tipo de gasolina Gasóleo A tanto para el de Gasolina 98, respectivamente.

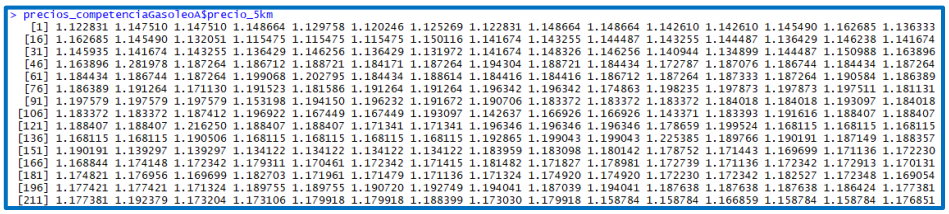


Figura 17: Precio medio de las gasolineras a 5 km para Gasóleo A

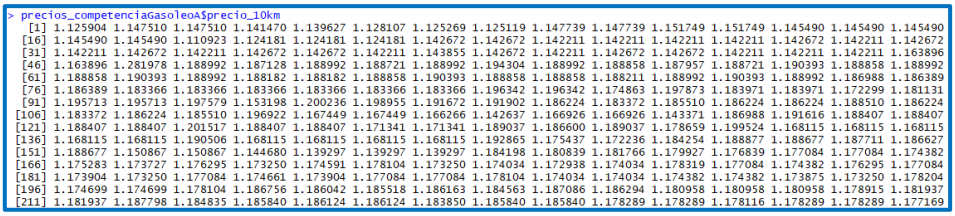


Figura 18: Precio medio de las gasolineras a 10 km para Gasóleo A

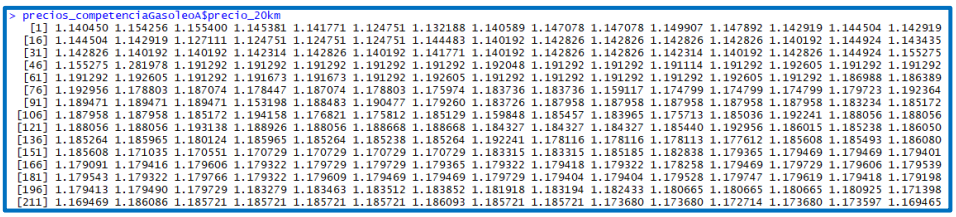


Figura 19: Precio medio de las gasolineras a 20 km para Gasóleo A

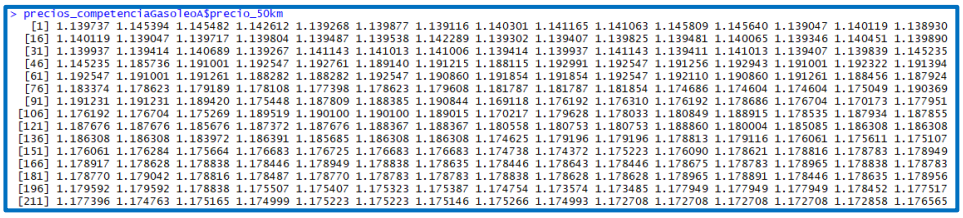


Figura 20: Precio medio de las gasolineras a 50 km para Gasóleo A

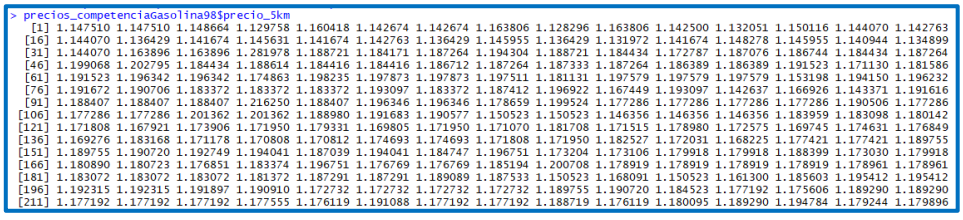


Figura 21: Precio medio de las gasolineras a 5 km para Gasolina 98

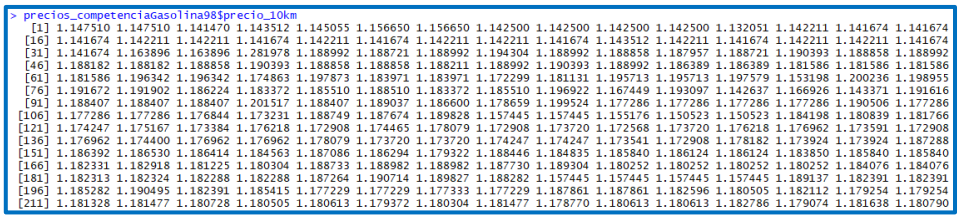


Figura 22: Precio medio de las gasolineras a 10 km para Gasolina 98

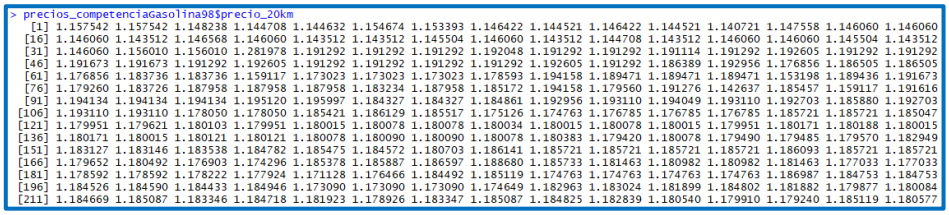


Figura 23: Precio medio de las gasolineras a 20 km para Gasolina 98

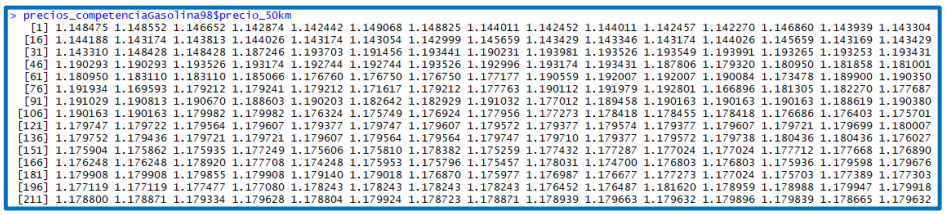


Figura 24: Precio medio de las gasolineras a 50 km para Gasolina 98

En la Figura 25 hasta la Figura 32, se puede observar una submuestra para cada una de las 8 variables creadas de los precios mínimos de las gasolineras a 5, 10, 20, y 50 km, “precio_5km”, “precio_10km”, “precio_20km”, y “precio_50km”, tanto para el tipo de gasolina Gasóleo A tanto para el de Gasolina 98, respectivamente.

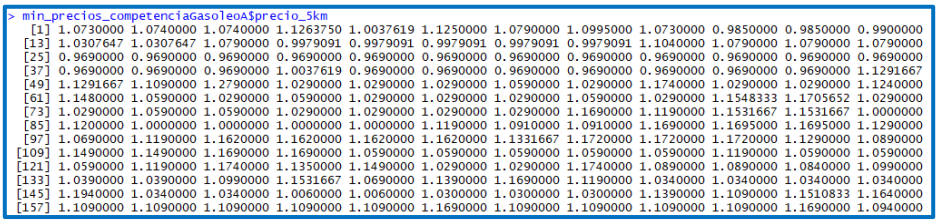


Figura 25: Precio mínimo de las gasolineras a 5 km para Gasóleo A

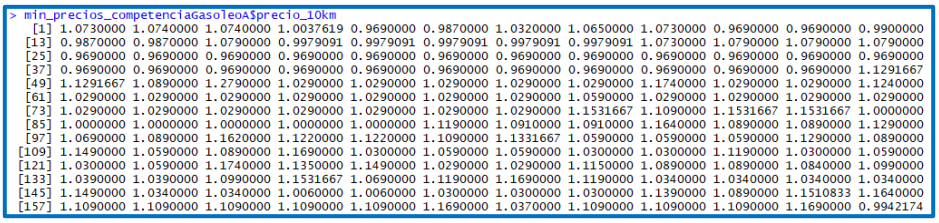


Figura 26: Precio mínimo de las gasolineras a 10 km para Gasóleo A

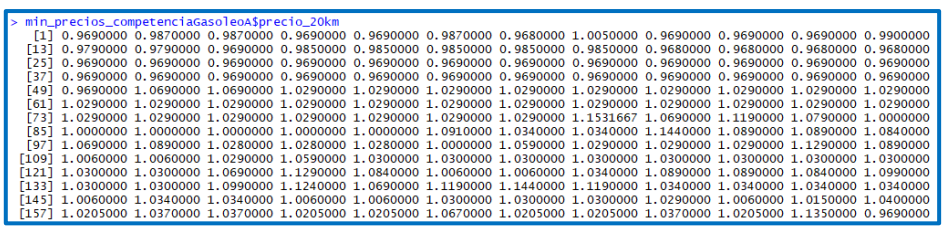


Figura 27: Precio mínimo de las gasolineras a 20 km para Gasóleo A

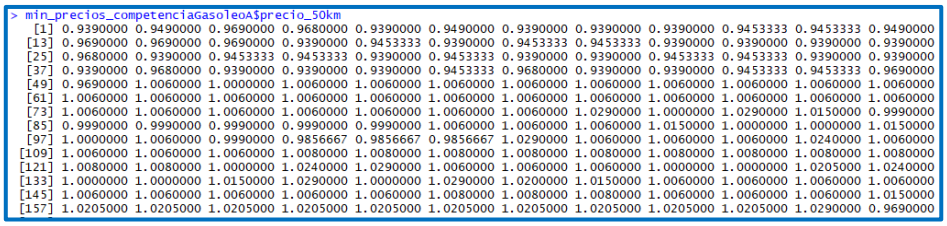


Figura 28: Precio mínimo de las gasolineras a 50 km para Gasóleo A

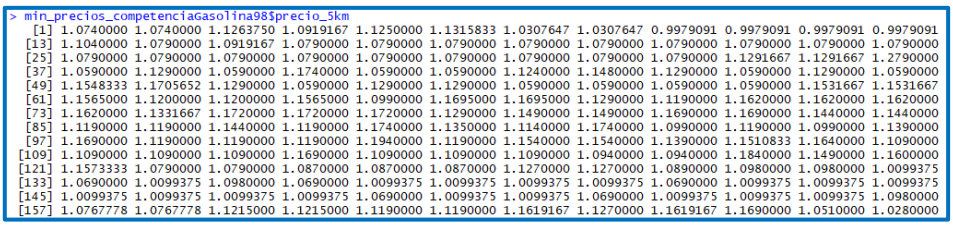


Figura 29: Precio mínimo de las gasolineras a 5 km para Gasolina 98

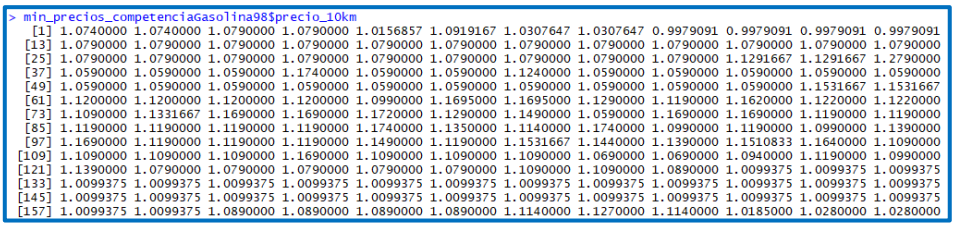


Figura 30: Precio mínimo de las gasolineras a 10 km para Gasolina 98

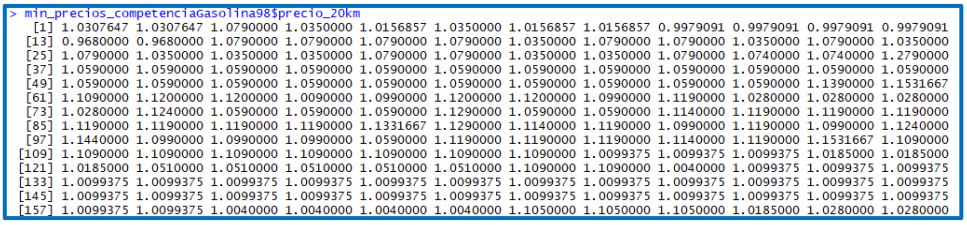


Figura 31: Precio mínimo de las gasolineras a 20 km para Gasolina 98

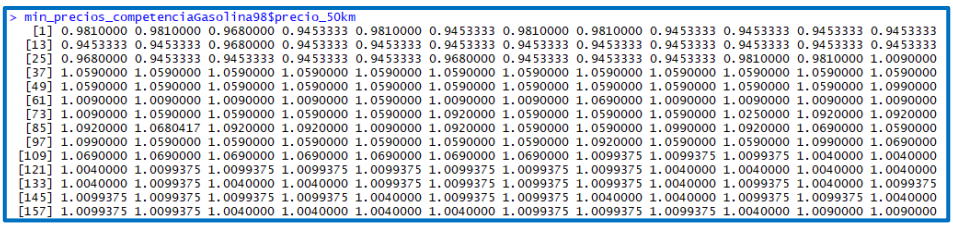


Figura 32: Precio mínimo de las gasolineras a 50 km para Gasolina 98

En el Anexo B podemos encontrar los gráficos y tablas correspondientes del análisis descriptivo, en las que se pueden observar que la media de las variables del precio medio para ambos tipos de gasolina, Gasóleo A y Gasolina 98 es muy parecido, en torno a 1.13, 1.14€ (lo cual también se puede observar en las Figura 17 hasta la Figura 24) (Anexo B, Tabla 1-2). Mientras que, el precio mínimo para ambos tipos de gasolina es 1€, aprox., (lo cual también se puede observar en las Figura 25 hasta la Figura 32) (Anexo B, Tabla 3-4).

Además, podemos observar que, tanto para los distintos tipos de gasolina como para el precio medio y mínimo, el número de gasolineras a medida que el radio aumenta es menor, lo cual es lógico, a pesar de que a 50 km hay zonas donde el número de gasolineras no se distribuye de forma exponencial (Anexo B, Figura 33-34) (Anexo B, Figura 35-36)

Posteriormente, para el tipo de gasolina Gasóleo A, realizaremos dos análisis clúster, uno referido para las 4 variables de los precios medios, y otro para las 4 variables de los precios mínimos.

7 Depuración de datos

Una vez realizado el análisis descriptivo de todas las variables objeto del estudio, vamos a llevar a cabo la depuración de los datos, como, la recodificación de las categorías de alguna de las variables, y el análisis de missing y atípicos.

Añadir, que, al disponer de dos conjuntos de datos, "precio_gasol_def" (precios de los carburantes con las variables dinámicas), y "gasolineras" (gasolineras), esta fase la tenemos que realizar para ambos ficheros.

7.1 Recodificación de las categorías de alguna de las variables

Como bien comentamos anteriormente en el capítulo anterior, en el análisis descriptivo, vamos a recodificar las categorías de la variable "horario", y de "rotulo", ya que disponemos de demasiados niveles, los cuales se pueden unir en categorías parecidas.

- En el caso de "horario", vamos a recategorizarla en 2 niveles, "abierto las 24h" y "no abre las 24h".

A continuación, se muestra la nueva recodificación de "horario" para ambos conjuntos de datos (Figura 37-38, respectivamente).

```
> precios_gasol_def$horario
  [1] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "abierto las 24 horas" "abierto las 24 horas"
  [6] "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "no abre las 24 horas"
 [11] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [16] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [21] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [26] "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas"
 [31] "abierto las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [36] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [41] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [46] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [51] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "abierto las 24 horas"
```

Figura 37: Recodificación de la variable "horario" en "precios_gasol_def"

```
> gasolineras$horario
  [1] "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas"
  [6] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [11] "no abre las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas"
 [16] "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas"
 [21] "abierto las 24 horas" "abierto las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [26] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "abierto las 24 horas"
 [31] "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas"
 [36] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [41] "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas" "no abre las 24 horas"
 [46] "no abre las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas"
 [51] "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas" "abierto las 24 horas"
```

Figura 38: Recodificación de la variable "horario" en "gasolineras"

- En el caso de "rotulo", vamos a recategorizarla en 9 niveles, categorías más frecuentes (REPSOL, CEPSA, GALP, SHELL, BP, PETRONOR, y CAMPSA), "supermercado" y "otros".

En la Figura 39 y 40, se muestra la nueva recodificación de "rotulo" en ambos conjuntos de datos.

```
> precios_gasol_def$rotulo
  [1] PETRONOR PETRONOR PETRONOR CAMPSA  CAMPSA  CAMPSA  SHELL   SHELL   SHELL   SHELL   SHELL   SHELL   CEPSA   CEPSA
 [15] CEPSA    REPSOL   REPSOL   REPSOL  REPSOL  REPSOL  REPSOL  otros   otros   otros   otros   otros   otros   otros
 [29] GALP     GALP     GALP     otros   otros   otros   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   otros   otros
 [43] otros    otros    otros    otros   REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  SHELL   SHELL   REPSOL  REPSOL
 [57] REPSOL   otros    otros    otros   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA
 [71] CEPSA    CEPSA    CEPSA    CEPSA   otros   otros   otros   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   REPSOL
 [85] REPSOL   REPSOL   otros    otros   otros   CEPSA   CEPSA   CEPSA   CEPSA   REPSOL  REPSOL  REPSOL  otros   otros
 [99] otros    otros    otros    otros   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   otros
[113] otros    otros    CEPSA    CEPSA   CEPSA   BP      BP      BP      SHELL   SHELL   SHELL   SHELL   SHELL   CEPSA
[127] CEPSA    CEPSA    CEPSA    GALP    GALP    GALP    GALP    GALP    GALP    CEPSA   CEPSA   CEPSA   CEPSA   REPSOL
[141] REPSOL   REPSOL   otros    otros   otros   otros   otros   otros   otros   GALP    GALP    GALP    otros   otros
[155] otros    otros    otros    otros   CEPSA   CEPSA   CEPSA   SHELL   SHELL   SHELL   CEPSA   CEPSA   CEPSA   otros
[169] otros    otros    otros    REPSOL  REPSOL  REPSOL  otros   otros   otros   otros   GALP    GALP    GALP    REPSOL
[183] REPSOL   REPSOL   REPSOL   otros   otros   otros   otros   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA
[197] otros    otros    otros    otros   otros   otros   REPSOL  REPSOL  REPSOL  SHELL   SHELL   otros   otros   otros
```

Figura 39: Recodificación de la variable "rotulo" en "precios_gasol_def"

```
> gasolineras$rotulo
  [1] REPSOL   REPSOL   otros    otros   otros   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   CEPSA   otros   otros   otros
 [15] otros    otros    otros    otros   otros   otros   CEPSA   CEPSA   PETRONOR PETRONOR PETRONOR REPSOL  REPSOL  REPSOL
 [29] REPSOL   otros    otros    otros   otros   otros   otros   otros   otros   PETRONOR PETRONOR PETRONOR otros   otros
 [43] otros    otros    otros    otros   SHELL   SHELL   PETRONOR PETRONOR PETRONOR otros   otros   otros   otros   PETRONOR
 [57] PETRONOR PETRONOR otros    otros   otros   GALP    GALP    GALP    GALP    otros   otros   otros   otros   otros
 [71] otros    otros    otros    otros   otros   otros   otros   otros   otros   otros   otros   otros   otros   otros
 [85] otros    otros    otros    CEPSA   CEPSA   CEPSA   otros   otros   otros   otros   otros   otros   otros   otros
 [99] otros    otros    otros    otros   otros   otros   otros   otros   otros   otros   otros   GALP    GALP    GALP
[113] SHELL    SHELL    SHELL    REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  PETRONOR
[127] PETRONOR PETRONOR PETRONOR PETRONOR PETRONOR REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL
[141] REPSOL   CEPSA    CEPSA    CEPSA   CEPSA   CEPSA   CEPSA   PETRONOR PETRONOR PETRONOR PETRONOR REPSOL  REPSOL  REPSOL
[155] otros    otros    otros    REPSOL  REPSOL  REPSOL  REPSOL  otros   otros   REPSOL  REPSOL  REPSOL  CEPSA   CEPSA
[169] CEPSA    otros    otros    otros   REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  REPSOL  CEPSA   CEPSA   CEPSA
[183] otros    otros    otros    otros   SHELL   SHELL   SHELL   SHELL   otros   otros   otros   otros   otros   otros
[197] otros    otros    otros    otros   otros   otros   otros   otros   otros   otros   otros   otros   REPSOL  REPSOL
```

Figura 40: Recodificación de la variable "rotulo" en "gasolineras"

7.2 Tratamiento de datos faltantes o missing

A continuación, vamos a observar si en alguna de las variables hay datos faltantes en alguno de los bastidores de datos.

En la Figura 41, podemos observar que para el conjunto de datos "gasolineras" existen 2 valores missing en las variables "latitud" y "longitud" (para "precios_gasol_def" ocurre lo mismo).

```
> which(is.na(precios_gasol_def$fecha))
integer(0)
> which(is.na(gasolineras$cp))
integer(0)
> which(is.na(gasolineras$horario))
integer(0)
> which(is.na(gasolineras$latitud))
[1] 11490 11491
> which(is.na(gasolineras$longitud))
[1] 11490 11491
> which(is.na(gasolineras$localidad))
integer(0)
> which(is.na(gasolineras$margen))
integer(0)
> which(is.na(gasolineras$municipio))
integer(0)
> which(is.na(gasolineras$rotulo))
integer(0)
> which(is.na(gasolineras$provincia))
integer(0)
> which(is.na(precios_gasol_def$tipo_gasol))
integer(0)
> which(is.na(precios_gasol_def$media_precio))
integer(0)
> which(is.na(gasolineras$direccion))
integer(0)
```

Figura 41: Datos faltantes

Por lo que procedemos a investigar qué ha ocurrido con esos valores faltantes, y observamos que esos missing provienen de la gasolinera "CTRA.CHUCENA-HINOJOS (CRTA. A-481 km 0,2)" (Figura 42), que entendemos que por motivos del sitio web no fueron escritos o tuvimos algún problema en la descarga de datos. Por tanto, para ambos conjuntos de datos procedemos a imputar la latitud y la longitud de forma manual a través de Google Maps, para así no disponer de ningún dato faltante.

	direccion	cp	horario	latitud	longitud	localidad	margen	municipio	rotulo	provincia	tipo_gasol
17272	CTRA.CHUCENA-HINOJOS (CRTA. A-481 km 0,2)	21891	no abre las 24 horas	NA	NA	CHUCENA	D	Chucena	(sin rótulo)	HUELVA	gasoleoA
17273	CTRA.CHUCENA-HINOJOS (CRTA. A-481 km 0,2)	21891	no abre las 24 horas	NA	NA	CHUCENA	D	Chucena	(sin rótulo)	HUELVA	gasolina95Proteccion

Figura 42: Gasolinera donde proceden los datos faltantes

7.3 Tratamiento de datos atípicos

Por último, vamos a observar si existen datos atípicos en nuestros conjuntos de datos de las variables continuas, como son, "latitud", "longitud", y "precio" ("precio" solo será evaluada en "precios_gasol_def" ya que se trata de una variable dinámica, y es en el fichero donde se encuentra) en función del tipo de gasolina, que como mencionamos anteriormente, en esta última se podía observar a simple vista que existían algunos posibles atípicos, lo cual lo corroboraremos a continuación (Anexo A, Figura 3).

Vamos a usar el Test de Grubbs, donde contrastamos:

$$H_0: El\ valor\ x\ de\ la\ variable\ continua\ y\ no\ es\ un\ atípico$$
$$H_1: El\ valor\ x\ de\ la\ variable\ continua\ y\ es\ un\ atípico$$

- Para la latitud, a pesar de que obtenemos un $p - valor < 2.2e\text{-}16$, lo que indica que para cualquier nivel de significación el valor 27.751944 es un outlier (Figura 43), no se trataría de un atípico ya que ese valor corresponde a Santa Cruz de Tenerife (Figura 44).

```
> grubbs.test(gasolineras$latitud, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  gasolineras$latitud
G = 3.89530, U = 0.99943, p-value < 2.2e-16
alternative hypothesis: lowest value 27.751944 is an outlier
```

Figura 43: Test de Grubbs para la latitud

	direccion	cp	horario	latitud	longitud	localidad	margen	municipio	rotulo	provincia	tipo_gasol
29887	CALLE CRTRA GRAL TIGADAY, S/N	38911	no abre las 24 horas	27.75194	-18.01194	FRONTERA	D	Frontera	DISA FRONTERA	SANTA CRUZ DE TENERIFE	gasoleoA
29888	CALLE CRTRA GRAL TIGADAY, S/N	38911	no abre las 24 horas	27.75194	-18.01194	FRONTERA	D	Frontera	DISA FRONTERA	SANTA CRUZ DE TENERIFE	gasolina95Proteccion
29889	CALLE CRTRA GRAL TIGADAY, S/N	38911	no abre las 24 horas	27.75194	-18.01194	FRONTERA	D	Frontera	DISA FRONTERA	SANTA CRUZ DE TENERIFE	gasolina98

Figura 44: Lugar al que pertenece el atípico 27.751944 de la latitud

Para el bastidor de datos "precios_gasol_def" concluimos los mismos resultados.

- Para la longitud, a pesar de que obtenemos un $p - valor < 2.2e\text{-}16$, lo que indica que para cualquier nivel de significación el valor 18.011944 es un outlier (Figura 45), no se trataría de un atípico ya que ese valor corresponde a Santa Cruz de Tenerife (Figura 44).

```
> grubbs.test(gasolineras$longitud, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  gasolineras$longitud
G = 3.76490, U = 0.99947, p-value < 2.2e-16
alternative hypothesis: lowest value -18.011944 is an outlier
```

Figura 45: Test de Grubbs para la latitud

Para el bastidor de datos "precios_gasol_def" concluimos los mismos resultados.

- En el caso del precio, vamos a filtrar en el bastidor "precios_gasol_def" por tipo de gasolina. En la Figura 46, podemos observar que en los precios de Gasolina 98, Gasóleo A, GNL, GNC, GLP, Bioetanol, obtenemos atípicos, ya que su $p - valor$ es menor a cualquier nivel de significación. Esto es debido a que su precio es bajo en los 5 primeros tipos de gasolina, mientras que para Bioetanol es porque su precio es alto (Figura 46). Por lo que vamos a realizar un histograma de estos tipos de gasolina, por separado, en función del precio para acotar los valores, y poder observar de una manera más visual si se trata de un atípico.

  A continuación, se muestran los histogramas de los precios de Gasolina 98, y de Gas Natural Licuado, en las Figuras 47-48, respectivamente.

  Para el precio de Gasolina 98, se puede observar en la Figura 47 que tenemos una función bimodal (distribución de probabilidad continua que posee dos modas diferentes, los cuales aparecen como picos distintos, máximos locales, en esta función) [31]. Por lo que el precio a 0.763€, el valor en el que el Test de Grubbs mostraba como atípico, concluimos que no se trataría de un atípico ya que los datos son bimodales.

  Para el precio de GNL, en la Figura 48 se puede observar que el precio a 0.674€ no se trataría de un atípico ya que se encuentra cerca de la moda. Además, se presentan grandes fluctuaciones porque hay una alta o baja competencia. Es decir, las gasolineras que tienen una baja competencia suben los precios, mientras que las que tienen mucha, los bajan. Esto es debido a que este tipo de gasolina no es tan común. Esto es lo que se llama la elasticidad de la demanda.

Para los demás tipos de gasolina, obtenemos las mismas conclusiones, ya que los histogramas son parecidos a los expuestos en la Figura 47-48, los cuales se pueden ver en el Anexo C.

Gasolina 98

```
> grubbs.test(precios_gasol_def[gasol98,]$media_precio, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  precios_gasol_def[gasol98, ]$media_precio
G = 7.12210, U = 0.99983, p-value = 3.196e-07
alternative hypothesis: lowest value 0.763 is an outlier
```

Gasolina 95 Protección

```
> grubbs.test(precios_gasol_def[gasol95Protec,]$media_precio, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  precios_gasol_def[gasol95Protec, ]$media_precio
G = 5.85890, U = 0.99992, p-value = 0.002013
alternative hypothesis: lowest value 0.799 is an outlier
```

Gasóleo B

```
> grubbs.test(precios_gasol_def[gasolB,]$media_precio, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  precios_gasol_def[gasolB, ]$media_precio
G = 4.86620, U = 0.99977, p-value = 0.1161
alternative hypothesis: highest value 1.1025 is an outlier
```

Gasóleo A

```
> grubbs.test(precios_gasol_def[gasolA,]$media_precio, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  precios_gasol_def[gasolA, ]$media_precio
G = 5.47200, U = 0.99993, p-value = 0.01997
alternative hypothesis: lowest value 0.739 is an outlier
```

Gas Natural Licuado

```
> grubbs.test(precios_gasol_def[gasNatLicuado,]$media_precio, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  precios_gasol_def[gasNatLicuado, ]$media_precio
G = 1.60700, U = 0.99722, p-value < 2.2e-16
alternative hypothesis: lowest value 0.674 is an outlier
```

Gas Natural Comprimido

```
> grubbs.test(precios_gasol_def[gasNatComprimido,]$media_precio, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  precios_gasol_def[gasNatComprimido, ]$media_precio
G = 2.11370, U = 0.99775, p-value < 2.2e-16
alternative hypothesis: lowest value 0.738 is an outlier
```

Gases Licuados Petróleo

```
> grubbs.test(precios_gasol_def[gasLicuadosPetrol,]$media_precio, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  precios_gasol_def[gasLicuadosPetrol, ]$media_precio
G = 4.0970, U = 0.9993, p-value = 0.9994
alternative hypothesis: lowest value 0.499 is an outlier
```

Bioetanol

```
> grubbs.test(precios_gasol_def[bioetan,]$media_precio, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  precios_gasol_def[bioetan, ]$media_precio
G = 2.74250, U = 0.98655, p-value < 2.2e-16
alternative hypothesis: highest value 1.699 is an outlier
```

Biodiésel

```
> grubbs.test(precios_gasol_def[biodios,]$media_precio, type = 10, opposite = FALSE, two.sided = TRUE)

        Grubbs test for one outlier

data:  precios_gasol_def[biodios, ]$media_precio
G = 4.31140, U = 0.99437, p-value = 0.05218
alternative hypothesis: highest value 1.318 is an outlier
```

Figura 46: Test de Grubbs para el precio en función del tipo de gasolina

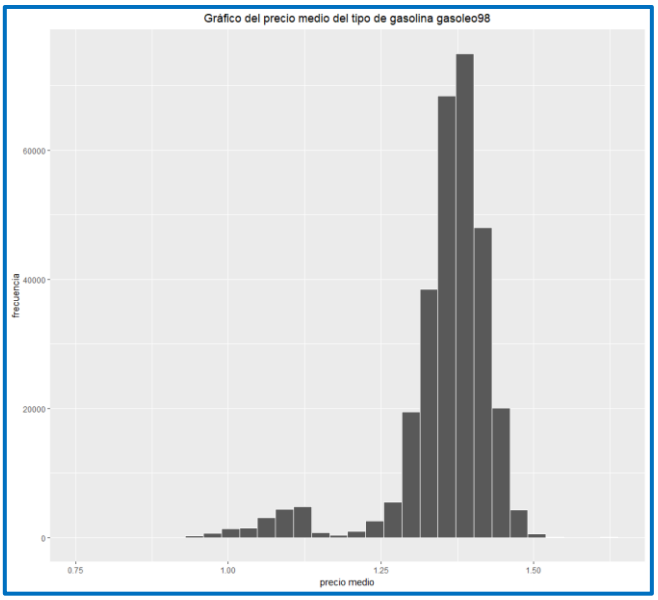


Figura 47: Histograma del precio medio de g98

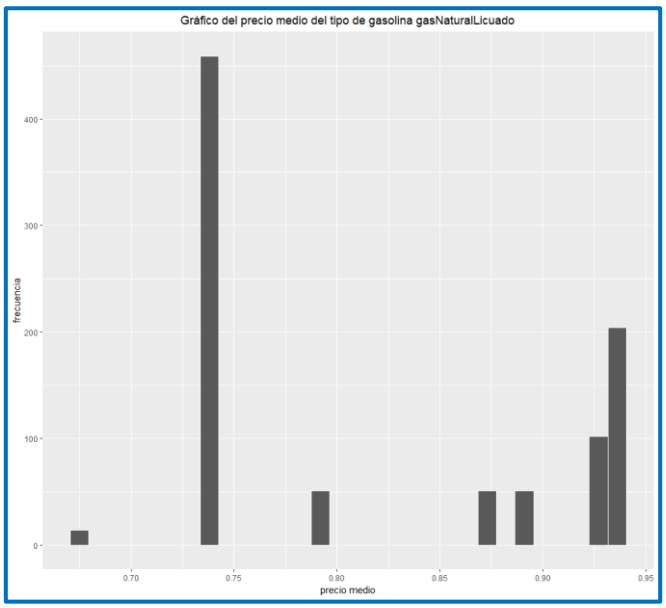


Figura 48: Histograma del precio medio de GNL

8 Variables finales

Una vez estudiadas y recategorizadas las variables originales se procede a presentar las variables finales, que serán las que utilicemos a lo largo del presente estudio, haciendo una distinción entre las variables propiamente descriptivas de una gasolinera, como son, la dirección, la latitud, la longitud, el código postal, su rótulo, entre otras; y variables de la competencia, referidas al precio medio y al precio mínimo de aquellas gasolineras que se encuentran a 5, 10, 20, y 50 km.

Añadir, que las únicas variables que han sido modificadas debido a la depuración fueron la variable de "horario", y de "rotulo" (recodificación de alguna de sus categorías), y la "latitud" y "longitud" ya que tenían 2 valores faltantes.

Por tanto, las variables finales van a ser las siguientes:

8.1 Variables descriptivas de una gasolinera

Las variables que describen a una gasolinera, objeto de estudio, son:

1) Fecha
2) Dirección
3) Código postal
4) Horario ("abierto las 24h", "no abre las 24h")
5) Latitud
6) Longitud
7) Localidad
8) Margen: posición en la que está situada la gasolinera (D: derecha, I: izquierda, N: a ambos lados)
9) Municipio
10) Rótulo: nombre de la gasolinera ("REPSOL", "CEPSA", "GALP", "SHELL", "BP", "PETRONOR", "CAMPSA", "Supermercados", "Otros".)
11) Provincia
12) Precio
13) Tipo gasolina (Gasóleo A, Gasóleo B, Gasolina 95 Protección, Gasolina 98, Biodiésel, Bioetanol, GLP, GNC, GNL)

8.2 Variables de la competencia

Las variables de la competencia para el precio medio son:

1) Precio medio a 5 km
2) Precio medio a 10 km
3) Precio medio a 20 km
4) Precio medio a 50 km

Para el precio mínimo, son:

1) Precio mínimo a 5 km
2) Precio mínimo a 10 km
3) Precio mínimo a 20 km
4) Precio mínimo a 50 km

9 Análisis multivariante

En este apartado, se clasificarán las gasolineras de España, atendiendo a las semejanzas y diferencias de perfiles existentes entre los comportamientos de la población. Para ello, se utilizará una técnica multivariante llamada Análisis Clúster.

9.1 Análisis Clúster

A continuación, vamos a proceder a realizar dos análisis clúster mediante el cual se clasificarán las gasolineras de España en distintos grupos, atendiendo a las semejanzas y diferencias de perfiles existentes. En el primer clúster, para crear estos grupos, nos basaremos en el precio de la gasolina, y del precio medio de la competencia a 5, 10, 20, y 50 km referentes al tipo de gasolina Gasóleo A, mientras que para el segundo y último clúster, para crear los grupos nos basaremos en el precio de la gasolina, y del precio mínimo de la competencia a 5, 10, 20, y 50 km referentes al tipo de gasolina Gasóleo A (cabe destacar que, estas variables fueron construidas para dos tipos de gasolina, Gasóleo A y Gasolina 98, y en este caso hemos plasmado los resultados con este carburante siendo similar para el otro).
Así, la competencia de cada gasolinera quedará establecida por los establecimientos que formen parte de su mismo grupo. Por lo que el objetivo perseguido en este punto es conseguir clasificar las distintas observaciones en grupos que tengan las siguientes propiedades:

- Cada grupo debe ser homogéneo con respecto a las variables utilizadas para su formación.
- Los grupos deben ser lo más distintos posible unos de otros.

El proceso que hemos llevado a cabo en este estudio se basa en:

- Un gráfico de la suma de cuadrados dentro de cada grupo por número de conglomerados extraídos por el que determinaremos el número apropiado de grupos o clúster.
- Un Clúster no Jerárquico. Una vez conocido el número concreto de grupos a crear, procedemos a llevar a cabo un proceso de clúster, en este caso, no jerárquico, llamado K-medias. Añadir, que este algoritmo sólo trata de encontrar patrones en los datos, y no da un resultado que predecir.

Añadir, que para el clúster 1, escogemos los precios medios de la competencia a 5, 10, 20, y 50 km, mientras que para el clúster 2, los precios mínimos, para así, tener dos focos de comparación y ver si hay diferencias en su interpretación.

Por último, para cada clúster, el del precio medio, y el del mínimo, obtendremos el número de gasolineras que hay en cada grupo, y un gráfico de las mismas por provincia y por conglomerado para así observar si en todos los grupos hay las mismas provincias.

Clúster 1: Precio de la gasolina, y precio medio de la competencia a 5, 10, 20, y 50 km

En primer lugar, tenemos que decidir el número óptimo de grupos en el gráfico de la suma de cuadrados dentro de cada grupo por número de conglomerados. Para decidir este número, tenemos que observar ese gráfico y en el punto en el que se estabilice esa curva es el número óptimo de conglomerados. En este caso, podemos observar que la curva se estabiliza con 4 clúster (Figura 57)

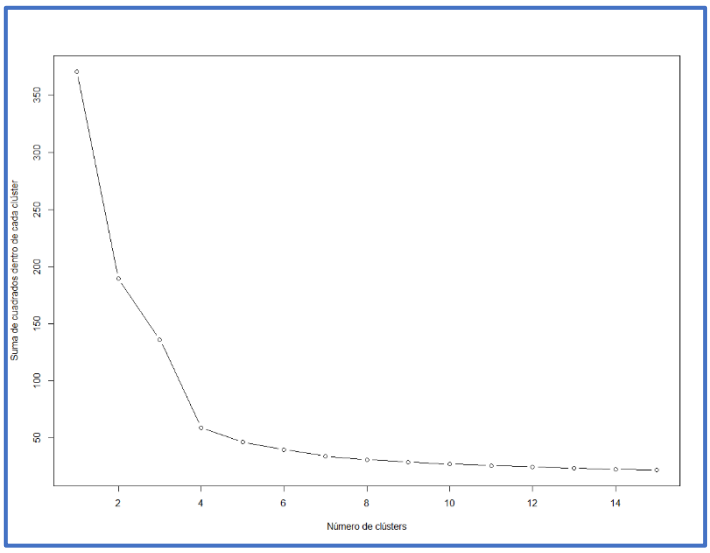


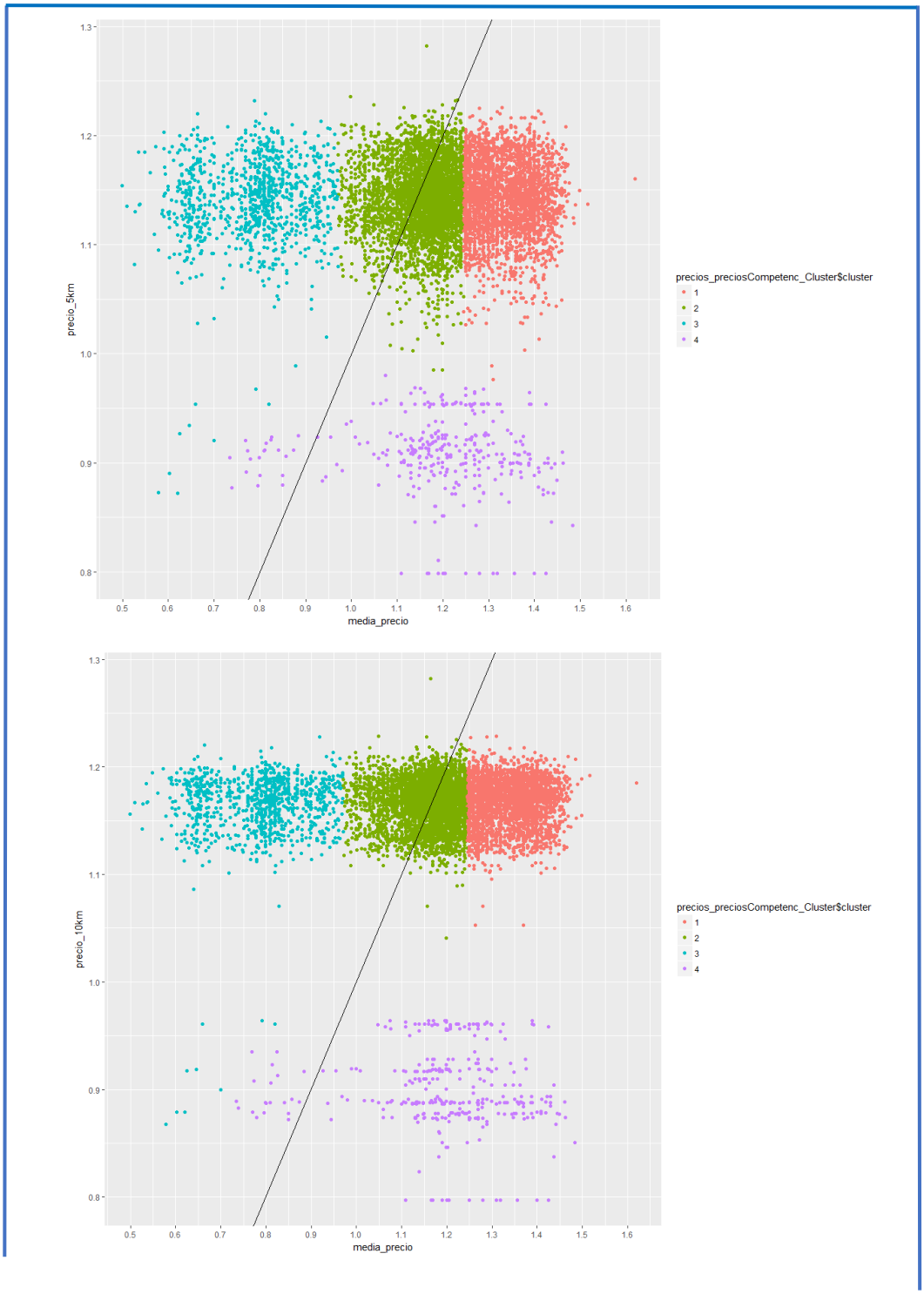
Figura 57: Gráfico del número óptimo de clúster en el Clúster 1

Por tanto, la agrupación de las gasolineras de España en 4 clústeres se muestra en la Figura 58, en el que se puede observar que el clúster 1 (C1) pertenece al color rojo, el 2 (C2) al color verde, el clúster 3 (C3) al azul, y, por último, el 4 (C4) al morado, cuya interpretación de cada uno de estos grupos es:

C1 = precio alto, competencia media. Se refiere a aquellas gasolineras en las que el precio es mayor que el de su competencia.
C2 = precio medio, competencia media. Se refiere a aquellas gasolineras en las que el precio es similar al de su competencia.
C3 = precio bajo, competencia media. Se refiere a aquellas gasolineras en las que el precio es menor que el de su competencia.
C4 = precio indiferente, competencia baja. Se refiere a aquellas gasolineras en las que el precio del entorno es bajo indiferentemente del precio de la propia gasolinera.

Añadir que, en el C2, C3, y C4, es donde los puntos se aglutinan más, lo que corresponde a las grandes gasolineras, como REPSOL, CAMPSA, entre otras, mientras que en el C1 es donde se aglutinan las pequeñas gasolineras, como podrían ser las de supermercado, como CARREFOUR, EROSKI, entre otras.

Además, podemos observar, que el precio del carburante no depende del precio medio de la competencia a 5 km, ni a 10 km, ni a 20 km, ni a 50 km, ya que la distribución de puntos es bastante parecida en todos los gráficos.


precio_5km
media_precio
precios_preciosCompetenc_Cluster$cluster
1
2
3
4
precio_10km
media_precio
precios_preciosCompetenc_Cluster$cluster
1
2
3
4

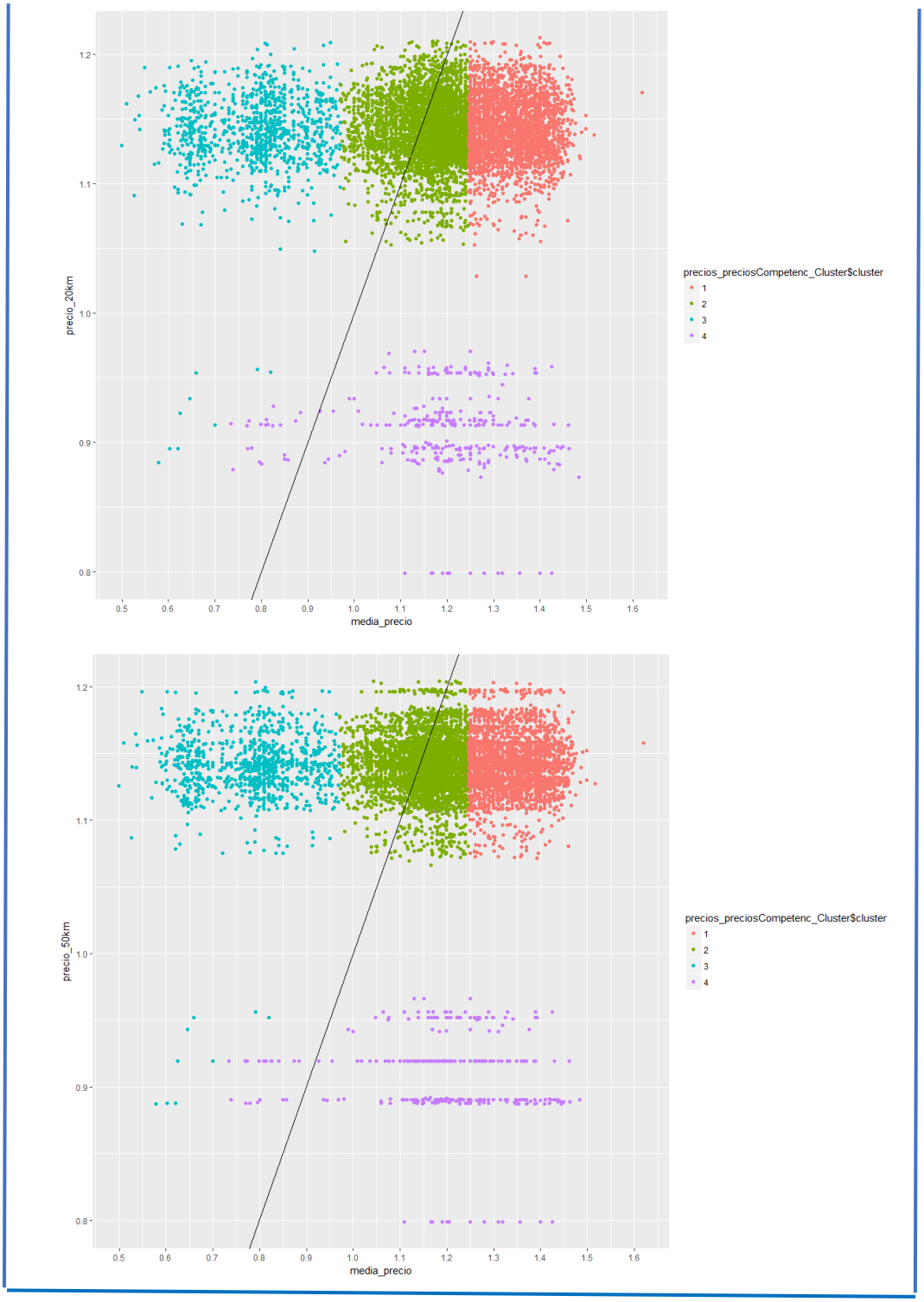


Figura 58: Clúster 1 de los precios de la gasolina y medios de la competencia

Por último, vamos a mostrar el número de gasolineras que hay en cada clúster (Figura 59), donde vemos que el C2 y C3 poseen aproximadamente el mismo, y, además, un gráfico de las mismas por provincia y por conglomerado (Figura 60), en el que observamos que en los 4 grupos forman parte las mismas provincias, siendo mayormente, en Madrid, Barcelona y Valencia.

```
> precios_preciosCompetenc_Cluster$size
[1]  347 3290 4281 1099
```

Figura 59: Número de gasolineras por clúster del Clúster 1

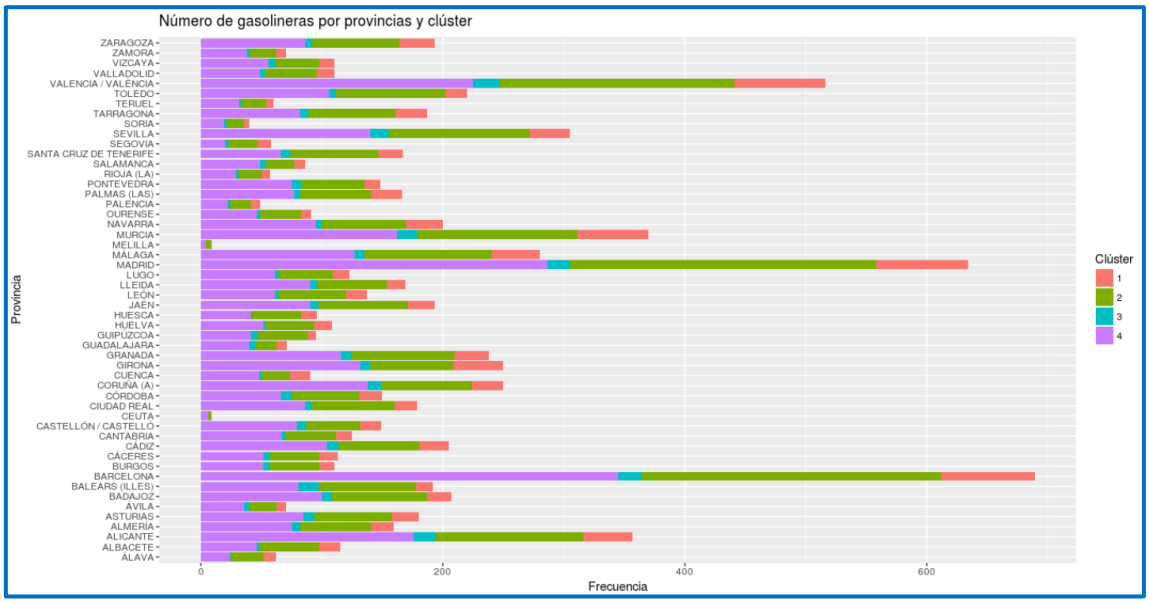


Figura 60: Gráfico del número de gasolineras por provincias y clúster del Clúster 1

B. Clúster 2: Precio de la gasolina, y precio mínimo de la competencia a 5, 10, 20, y 50 km

Se obtienen resultados parecidos, los cuales se pueden observar en el Anexo D.

10 Modelos predictivos

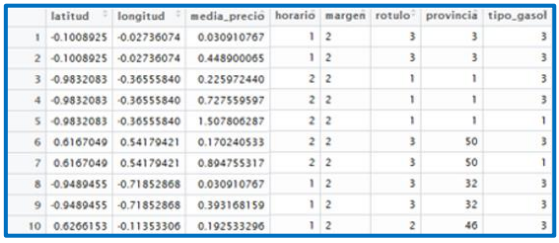
Una vez alcanzados los apartados anteriores llegamos al punto donde afrontaremos uno de los objetivos del presente estudio, predecir el precio del carburante de las gasolineras de España, en función de una serie de variables generales de estas, como son, el nombre, y, el tipo de gasolina, entre otras.

Para abordar este objetivo se utilizarán 4 técnicas estadísticas de predicción, la Regresión Lineal, Redes Neuronales, Random Forest, y Suport Vector Machine, para los cuales se proporcionarán medidas de ajuste y comparativas entre ambos para decidir cuál de los modelos conseguidos es el que mejor resuelve este objetivo.
Además, vamos a prescindir de aquellas variables cualitativas que poseen muchas categorías ya que al realizar los distintos modelos podemos obtener un problema de sobreparametrización. En este caso, vamos a eliminar a priori la localidad, el municipio, la dirección y el código postal. Por último, hemos estandarizado solo las variables continuas para que sea más fácil la convergencia en el caso de Redes, y creado variables dummys para las variables categóricas, para su incorporación en el modelo.

10.1 División del conjunto de datos

Se decide que la mejor forma de desarrollar estas técnicas de predicción, así como su posterior validación y pruebas, será dividiendo el conjunto de datos original en 2 subconjuntos de datos distintos, los cuales tendrán información relativa al 70% y 30% de los datos, respectivamente (hemos escogido esos porcentajes debido a que son los más usuales en minería de datos). Es decir,

- *Entrenamiento*: este conjunto de datos abarcará el 70% de los precios, ubicaciones, y demás aspectos de las distintas gasolineras de España, el cual será utilizado por cada uno de los modelos de predicción que se van a llevar a cabo para predecir los valores de los parámetros inherentes a cada modelo. Consta de 910706 observaciones. Un subconjunto de este se puede ver en la Figura 64.

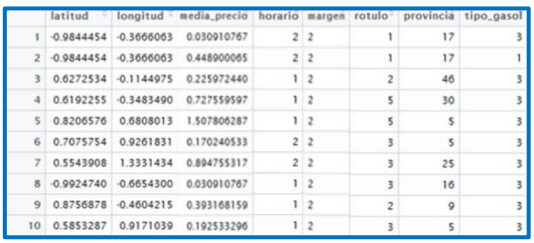


	latitud	longitud	media_precio	horario	margen	rotulo	provincia	tipo_gasol
1	-0.1008925	-0.02736074	0.030910767	1	2	3	3	3
2	-0.1008925	-0.02736074	0.448900065	1	2	3	3	3
3	-0.9832083	-0.36555840	0.225972440	2	2	1	1	3
4	-0.9832083	-0.36555840	0.727559597	2	2	1	1	3
5	-0.9832083	-0.36555840	1.507806287	2	2	1	1	1
6	0.6167049	0.54179421	0.170240533	2	2	3	50	3
7	0.6167049	0.54179421	0.894755317	2	2	3	50	1
8	-0.9489455	-0.71852868	0.030910767	1	2	3	32	3
9	-0.9489455	-0.71852868	0.393168159	1	2	3	32	3
10	0.6266153	-0.11353306	0.192533296	1	2	2	46	3

Figura 64: Subconjunto del bastidor de Entrenamiento

- *Test*: conjunto de datos relativo al 30% de los precios, ubicaciones, y demás aspectos de las distintas gasolineras de España. Dicha información será utilizada para predecir de forma insesgada el grado de acierto de los modelos seleccionados como mejores en validación cruzada repetida. Consta de 303545 observaciones. Un subconjunto de este se puede ver en la Figura 65.

Asimismo, se ha utilizado la técnica de validación cruzada repetida con el objetivo de decidir cuál de todos los modelos de predicción llevados a cabo comete un error menor en la estimación, siendo así el mejor, en vez de dividir el bastidor de datos en 3 subconjuntos (Entrenamiento, Validación, y Test).



	latitud	longitud	media_precio	horario	margen	rotulo	provincia	tipo_gasol
1	-0.9844454	-0.3666063	0.030910767	2	2	1	17	3
2	-0.9844454	-0.3666063	0.448900065	2	2	1	17	1
3	0.6272534	-0.1144975	0.225972440	1	2	2	46	3
4	0.6192255	-0.3483490	0.727559597	1	2	5	30	3
5	0.8206576	0.6808013	1.507806287	1	2	5	5	3
6	0.7075754	0.9261831	0.170240533	2	2	3	5	3
7	0.5543908	1.3331434	0.894755317	2	2	3	25	3
8	-0.9924740	-0.6654300	0.030910767	1	2	3	16	3
9	0.8756878	-0.4604215	0.393168159	1	2	2	9	3
10	0.5853287	0.9171039	0.192533296	1	2	3	5	3

Figura 65: Subconjunto del bastidor de Test

10.2 Regresión lineal

- **Construcción del modelo**

  La construcción del mejor modelo de Regresión Lineal se lleva a cabo variando los diferentes parámetros propios de esta técnica de forma manual y lógica, como

son, el método de selección de variables (Stepwise, Forward, y Backward), el número de grupos en validación cruzada, y la semilla, respectivamente. Esto se llevará a cabo sobre el conjunto de datos de Entrenamiento, para después valorar su funcionamiento sobre la técnica de validación cruzada repetida.

En este caso, vamos a usar para todos los modelos, 5 como número de semillas diferentes ya que creemos que es un número óptimo para la obtención del error medio de predicción. Además, las variables seleccionadas en cada método de selección de variables son las mismas que introduciendo todas las variables del conjunto de Entrenamiento. Es decir,

LATITUD, LONGITUD, HORARIO, ROTULO, PROVINCIA, TIPO_GASOLINA (Figura 66)

```
> modelo <- lm(media_precio~.-1, data = precios2)
> modelo

Call:
lm(formula = media_precio ~ . - 1, data = precios2)

Coefficients:
                          latitud                         longitud                      rotuloSHELL
                         0.108932                         0.080597                        -0.364782
     horarioabierto las 24 horas                     rotuloCAMPSA                  provinciaBURGOS
                        -0.016533                        -0.212450                        -0.107198
                provinciaMÁLAGA                  provinciaZARAGOZA                    provinciaJAÉN
                        -0.576125                        -0.286791                        -0.150876
             provinciaBARCELONA                     provinciaLEÓN                  provinciaSEVILLA
                        -0.218812                        -0.208669                        -0.356807
               provinciaGRANADA     provinciaVALENCIA / VALÈNCIA                  provinciaTOLEDO
                        -0.437897                        -0.318142                        -0.810014
                provinciaHUELVA                  provinciaBADAJOZ                  provinciaPALENCIA
                         0.005266                        -0.191921                         0.025411
                provinciaZAMORA                   provinciaMADRID                   provinciaCUENCA
                        -0.538427                        -0.401557                        -0.084387
              provinciaTARRAGONA                   provinciaNAVARRA                    provinciaÁVILA
                        -0.143163                        -0.132610                        -0.105219
           provinciaGUADALAJARA                   provinciaLLEIDA     tipo_gasolgasolina95Proteccion
                        -0.252841                        -0.106213                         0.723940
              provinciaALBACETE                    provinciaLUGO   tipo_gasolgasesLicuadosPetroleo
                        -0.069286                        -0.483436                         0.286198
          tipo_gasolgasolina98                tipo_gasolgasoleoB
                        -2.364867                         0.827688
```

Figura 66: Variables seleccionadas en RLineal

Se han propuesto 10 modelos, de los cuales los 5 primeros corresponden a la semilla 12346, y los 5 restantes se refieren a otra semilla, 12349, para observar si estos modelos presentan el mismo comportamiento a pesar de utilizar una semilla distinta. Dichos modelos se muestran en la Figura 67.

```
lin <- cvrepetidalin(precios2,4,"media_precio",12346,5)
lin1 <- cvrepetidalin(precios2,5,"media_precio",12346,5)
lin2 <- cvrepetidalin(precios2,3,"media_precio",12346,5)
lin3 <- cvrepetidalin(precios2,2,"media_precio",12346,5)
lin4 <- cvrepetidalin(precios2,6,"media_precio",12346,5)
lin5 <- cvrepetidalin(precios2,4,"media_precio",12349,5) #distinta semilla
lin6 <- cvrepetidalin(precios2,5,"media_precio",12349,5)
lin7 <- cvrepetidalin(precios2,3,"media_precio",12349,5)
lin8 <- cvrepetidalin(precios2,2,"media_precio",12349,5)
lin9 <- cvrepetidalin(precios2,6,"media_precio",12349,5)
```

Figura 67: Modelos propuestos de RLineal

- **Comparación de modelos**

Una vez propuestos los modelos anteriores, vamos a proceder a la comparación de estos. Para ello, vamos a obtener un gráfico de caja en el que se muestra cada uno de los modelos frente a su error medio de predicción en validación cruzada

repetida, en el que podemos observar que, para la semilla 12346, todos los modelos son igual de buenos ($lin - lin4$) ya que obtenemos un error similar, en torno a 0.17. Por lo que concluimos que el modelo óptimo de regresión lineal es el segundo, $lin1$, que es con el que obtenemos un menor error (Figura 68), seleccionando la mayoría de las variables como significativas (excepto "horarioabierto 24 horas", "provinciaCUENCA"), ya que para cualquier nivel de significación obtenemos un $p - valor < 2e\text{-}16$ (Figura 69), cuyo contraste sería:

$$H_0: \beta_i = 0$$
$$H_1: \beta_i \neq 0$$

Añadir que para la semilla 12349 ($lin5 - lin9$), se comportan prácticamente de la misma forma, lo cual es lógico (Figura 68).

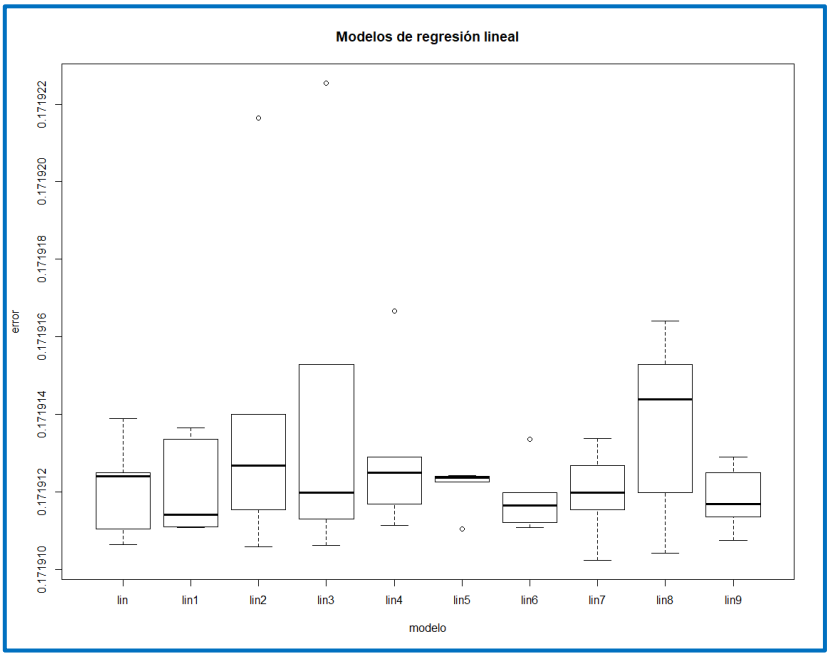


Figura 68: Comparación de los modelos de RLineal

Por lo que el modelo óptimo de regresión lineal $lin1$, queda expresado matemáticamente por la siguiente ecuación:

$$precio = 0.1250960 * latitud + 0.0471159 * longitud - 0.212450 * rotuloCAMPSA - 0.364782 * rotuloSHELL - 0.576125 * provinciaMÁLAGA - 0.286791 * provinciaZARAGOZA - 0.107198 * provinciaBURGOS - 0.218812 * provinciaBARCELONA - 0.208669 * provinciaLEÓN - 0.150876 * provinciaJAÉN - 0.437897 * provinciaGRANADA - 0.318142 * provinciaVALENCIA - 0.356807 * provinciaSEVILLA - 0.191921 * provinciaBADAJOZ - 0.810014 * provinciaTOLEDO - 0.538427 * provinciaZAMORA - 0.401557 * provinciaMADRID - 0.025411 * provinciaPALENCIA - 0.143163 * provinciaTARRAGONA - 0.132610 * provinciaNAVARRA - 0.084387 * provinciaMÁLAGA - 0.252841 * provinciaCUENCA - 0.106213 * provinciaGUADALAJARA - 0.105219 * provinciaLLEIDA - 0.060286 * provinciaÁVILA - 0.483436 * provinciaALBACETE - 0.423940 * provinciaLUGO + 0.723940 * tipo_gasolgasolina95Proteccion - 2.364867 * tipo_gasolgasolina98 + 0.827688 * tipo_gasolgasoleoA + 0.286198 * tipo_gasolgasesLicuadosPetroleo$$

```
> summary(modelo)

Call:
lm(formula = media_precio ~ . - 1, data = precios2)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6382 -0.2071  0.1038  0.4346  2.7857

Coefficients:
                                Estimate Std. Error  t value Pr(>|t|)
latitud                         0.108932   0.001278   85.245  < 2e-16 ***
longitud                        0.080597   0.001473   54.733  < 2e-16 ***
horarioabierto las 24 horas    -0.016533   0.018748   -0.882    0.378
rotuloCAMPSA                   -0.212450   0.006217  -34.174  < 2e-16 ***
rotuloSHELL                    -0.364782   0.006007  -60.726  < 2e-16 ***
provinciaMÁLAGA                -0.576125   0.006780  -84.980  < 2e-16 ***
provinciaZARAGOZA              -0.286791   0.006684  -42.909  < 2e-16 ***
provinciaBURGOS                -0.107198   0.003716  -28.844  < 2e-16 ***
provinciaBARCELONA             -0.218812   0.009082  -24.094  < 2e-16 ***
provinciaLEÓN                  -0.208669   0.004046  -51.575  < 2e-16 ***
provinciaJAÉN                  -0.150876   0.005375  -28.068  < 2e-16 ***
provinciaGRANADA               -0.437897   0.008108  -54.009  < 2e-16 ***
provinciaVALENCIA / VALÈNCIA   -0.318142   0.006691  -47.546  < 2e-16 ***
provinciaSEVILLA               -0.356807   0.005892  -60.556  < 2e-16 ***
provinciaHUELVA                 0.005266   0.005041    1.045    0.296
provinciaBADAJOZ               -0.191921   0.006711  -28.598  < 2e-16 ***
provinciaTOLEDO                -0.810014   0.006539 -123.875  < 2e-16 ***
provinciaZAMORA                -0.538427   0.011744  -45.848  < 2e-16 ***
provinciaMADRID                -0.401557   0.005346  -75.107  < 2e-16 ***
provinciaPALENCIA               0.025411   0.005384    4.720 2.36e-06 ***
provinciaTARRAGONA             -0.143163   0.007020  -20.394  < 2e-16 ***
provinciaNAVARRA               -0.132610   0.004424  -29.976  < 2e-16 ***
provinciaCUENCA                -0.084387   0.005889  -14.331  < 2e-16 ***
provinciaGUADALAJARA           -0.252841   0.007323  -34.525  < 2e-16 ***
provinciaLLEIDA                -0.106213   0.005049  -21.035  < 2e-16 ***
provinciaÁVILA                 -0.105219   0.005327  -19.751  < 2e-16 ***
provinciaALBACETE              -0.069286   0.004114  -16.843  < 2e-16 ***
provinciaLUGO                  -0.483436   0.006476  -74.645  < 2e-16 ***
tipo_gasolgasolina95Proteccion  0.723940   0.017603   41.125  < 2e-16 ***
tipo_gasolgasolina98           -2.364867   0.018546 -127.516  < 2e-16 ***
tipo_gasolgasoleoB              0.827688   0.017596   47.038  < 2e-16 ***
tipo_gasolgasesLicuadosPetroleo 0.286198   0.017593   16.268  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figura 69: Coeficientes del mejor modelo de RLineal

- **Interpretación del modelo óptimo**

  Por último, vamos a interpretar alguno de los coeficientes de este modelo, los cuales van a ser los de una variable continua, y otra cualitativa.

  LATITUD: por cada grado que nos desplacemos hacia la latitud, el precio de la gasolina aumentará 0.1089€.

  TIPO_GASOLINA: el precio del carburante Gasolina 95 Protección aumentará 0.72€ aproximadamente, respecto a Gasóleo A (variable de referencia).

  Las correspondientes salidas que no se muestran se pueden observar en el Anexo E.

10.3 Redes Neuronales

- **Construcción del modelo**

  La construcción del mejor modelo de Red Neuronal se lleva a cabo variando los diferentes parámetros propios de esta técnica de forma manual y lógica, como son, el número de grupos en validación cruzada, el número de nodos, número de iteraciones máximas permitidas, el algoritmo de activación (TANH, RECTIFICIERWITHDROPOUT, y MAXOUT), y la semilla, respectivamente. Esto se llevará a cabo sobre el conjunto de datos de Entrenamiento, para después valorar su funcionamiento sobre la técnica de validación cruzada repetida.

  En este caso, para todos los modelos vamos a utilizar 5 como número de semillas diferentes ya que creemos que es un número óptimo para la obtención del error medio de predicción, y debido a las limitaciones del paquete que hemos utilizado, BACKPROPAGATION como algoritmo de activación y 1 capa.

  Se han propuesto 14 modelos, de los cuales los 7 primeros corresponden a la semilla 12346, y los 7 restantes se refieren a otra semilla, 12349, para observar si estos modelos presentan el mismo comportamiento a pesar de utilizar una semilla distinta. Dichos modelos se muestran en la Figura 73.

```
red1 <- cvrepetidannet(precios2,4,"media_precio",12346, 5, 5, 10)
red2 <- cvrepetidah2o(precios2,4,"media_precio",12346,5, 8, 20, "Tanh")
red3 <- cvrepetidah2o(precios2,5,"media_precio",12346,5, 10, 30, "Tanh")
red4 <- cvrepetidah2o(precios2,3,"media_precio",12346,5, 12, 40, "TanhWithDropout")
red5 <- cvrepetidah2o(precios2,3,"media_precio",12346,5, 13, 50, "TanhWithDropout")
red6 <- cvrepetidah2o(precios2,6,"media_precio",12346,5, 15, 60, "Maxout")
red7 <- cvrepetidah2o(precios2,4,"media_precio",12346,5, 17, 70, "Maxout")
red8 <- cvrepetidannet(precios2,5,"media_precio",12349, 5, 5, 10) #distinta semilla
red9 <-  cvrepetidah2o(precios2,3,"media_precio",12349,5, 8, 20, "Tanh")
red10 <- cvrepetidah2o(precios2,3,"media_precio",12349,5, 10, 30, "Tanh")
red11 <- cvrepetidah2o(precios2,2,"media_precio",12349,5, 12, 40, "TanhWithDropout")
red12 <- cvrepetidah2o(precios2,6,"media_precio",12349,5, 13, 50, "TanhWithDropout")
red13 <- cvrepetidah2o(precios2,6,"media_precio",12349,5, 15, 60, "Maxout")
red14 <- cvrepetidah2o(precios2,6,"media_precio",12349,5, 17, 70, "Maxout")
```

Figura 73: Modelos propuestos de RN

- **Comparación de modelos**

  Una vez propuestos los modelos anteriores, vamos a proceder a la comparación de estos. Para ello, vamos a obtener un gráfico de caja en el que se muestra cada uno de los modelos frente a su error medio de predicción en validación cruzada repetida, en el que podemos observar en la Figura 74 que, para la semilla 12346 ($red1 - h2o7$), los modelos tienen el mismo comportamiento que con la semilla 12349 ($h2o8 - h2o14$), lo que era de esperar. Por lo que concluimos que el modelo óptimo de red neuronal es el séptimo, $h2o7$, ya que es el que consigue un menor error, aproximadamente 0.15.

  Por lo que los parámetros del modelo óptimo de red neuronal $h2o7$, son:

  - Variables a utilizar: las del mejor modelo de regresión lineal (LATITUD, LONGITUD, HORARIO, ROTULO, PROVINCIA, TIPO_GASOLINA).
  - Algoritmo de optimización: MAXOUT
  - Algoritmo de activación: BACKPROPAGATION
  - Número de nodos: 17
  - Número de capas: 1
  - Máximo número de iteraciones permitidas: 70

  En el Anexo F, podemos observar la correspondiente salida del gráfico de comparación de modelos de RN de la Figura 74 más ampliado y separados por valores de error similares, ya que hay modelos que quizás no se aprecian correctamente.

- **Interpretación del modelo óptimo**

  Por último, se procede a calcular la importancia de cada una de las variables independientes, que indica cuánto cambia el valor pronosticado por el modelo óptimo de Redes Neuronales para diferentes valores de la variable independiente. Dicha importancia se establecerá en función de la variabilidad del error cuadrado medio, la cual se puede muestra en la Figura 75.

  En la Figura 75, se puede observar que la variable con mayor importancia es "tipo_gasol", la cual corresponde al tipo de gasolina, aunque el resto de variables también son importantes para este modelo.

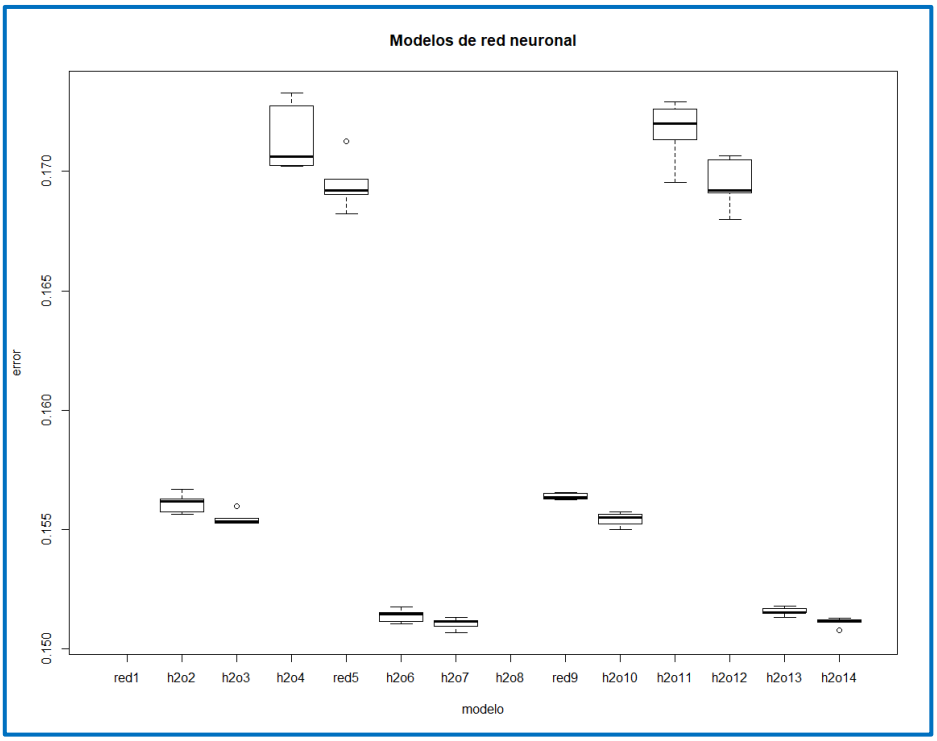


Figura 74: Comparación de modelos de RN

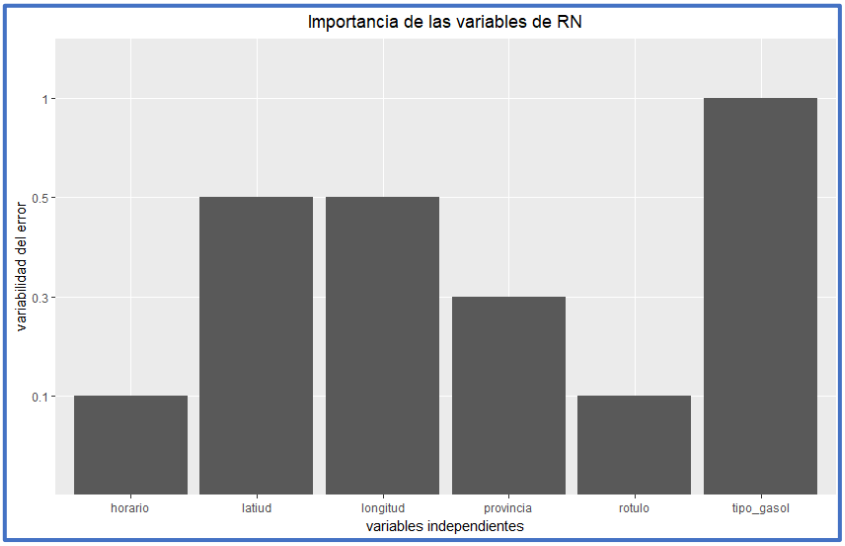


Figura 75: Importancia de las variables independientes de la RN óptima.

10.4 Random Forest

- **Construcción del modelo**

  La construcción del mejor modelo de Random Forest se lleva a cabo variando los diferentes parámetros propios de esta técnica de forma manual y lógica, como son, el número de grupos en validación cruzada, número de iteraciones máximas permitidas, número de observaciones por nodo, el tamaño de hoja mínimo (hasta 20), y la semilla, respectivamente. Esto se llevará a cabo sobre el conjunto de datos

de Entrenamiento, para después valorar su funcionamiento sobre la técnica de validación cruzada repetida.

En este caso, para todos los modelos vamos a utilizar 5 como número de semillas diferentes ya que creemos que es un número óptimo para la obtención del error medio de predicción, 7 como número de variables a tener en cuenta uno de los nodos de los diferentes árboles.

Se han propuesto 14 modelos, de los cuales los 7 primeros corresponden a la semilla 12346, y los 7 restantes se refieren a otra semilla, 12349, para observar si estos modelos presentan el mismo comportamiento a pesar de utilizar una semilla distinta. Dichos modelos se muestran en la Figura 78.

```
rf <- cvrepetidarf(precios2muestra,4,"media_precio",12346,5, 10, 5, 7)
rf1 <- cvrepetidarf(precios2muestra,3,"media_precio",12346,5, 20, 5, 7)
rf2 <- cvrepetidarf(precios2muestra,5,"media_precio",12346,5, 30, 5, 7)
rf3 <- cvrepetidarf(precios2muestra,6,"media_precio",12346,5, 40, 6, 7)
rf4 <- cvrepetidarf(precios2muestra,3,"media_precio",12346,5, 50, 2, 7)
rf5 <- cvrepetidarf(precios2muestra,4,"media_precio",12346,5, 60, 3, 7)
rf6 <- cvrepetidarf(precios2muestra,4,"media_precio",12346,5, 70, 3, 7)
rf7 <- cvrepetidarf(precios2muestra,4,"media_precio",12349,5, 10, 5, 7) #distinta semilla
rf8 <- cvrepetidarf(precios2muestra,3,"media_precio",12349,5, 20, 5, 7)
rf9 <- cvrepetidarf(precios2muestra,5,"media_precio",12349,5, 30, 5, 7)
rf10 <- cvrepetidarf(precios2muestra,6,"media_precio",12349,5, 40, 6, 7)
rf11 <- cvrepetidarf(precios2muestra,3,"media_precio",12349,5, 50, 2, 7)
rf12 <- cvrepetidarf(precios2muestra,4,"media_precio",12349,5, 60, 3, 7)
rf13 <- cvrepetidarf(precios2muestra,4,"media_precio",12349,5, 70, 3, 7)
```

Figura 78: Modelos propuestos de RF

- **Comparación de modelos**

Una vez propuestos los modelos anteriores, vamos a proceder a la comparación de estos. Para ello, vamos a obtener un gráfico de caja en el que se muestra cada uno de los modelos frente a su error medio de predicción en validación cruzada repetida, en el que podemos observar en la Figura 79 que, para la semilla 12346 (rf – $rf6$), los modelos tienen el mismo comportamiento que con la semilla 12349 ($rf7$– $fr13$), lo que era de esperar. En este caso, todos los modelos son igual de óptimos ya que con todos ellos se consigue un error de 0.17 aproximadamente. Por lo que concluimos que el modelo óptimo de red neuronal es el segundo, $rf1$, ya que es el que consigue un menor error.

Por lo que los parámetros del modelo óptimo de random forest $rf1$, son:

- Variables a utilizar: las del mejor modelo de regresión lineal (LATITUD, LONGITUD, HORARIO, ROTULO, PROVINCIA, TIPO_GASOLINA).
- Máximo número de iteraciones permitidas: 20
- Número de observaciones por nodo: 5
- Número de variables: 7
- Tamaño de hoja mínimo: 20

En el Anexo G, podemos observar la correspondiente salida del gráfico de comparación de modelos de RF de la Figura 79 más ampliado y separados por comportamientos similares, ya que hay modelos que quizás no se aprecian correctamente.

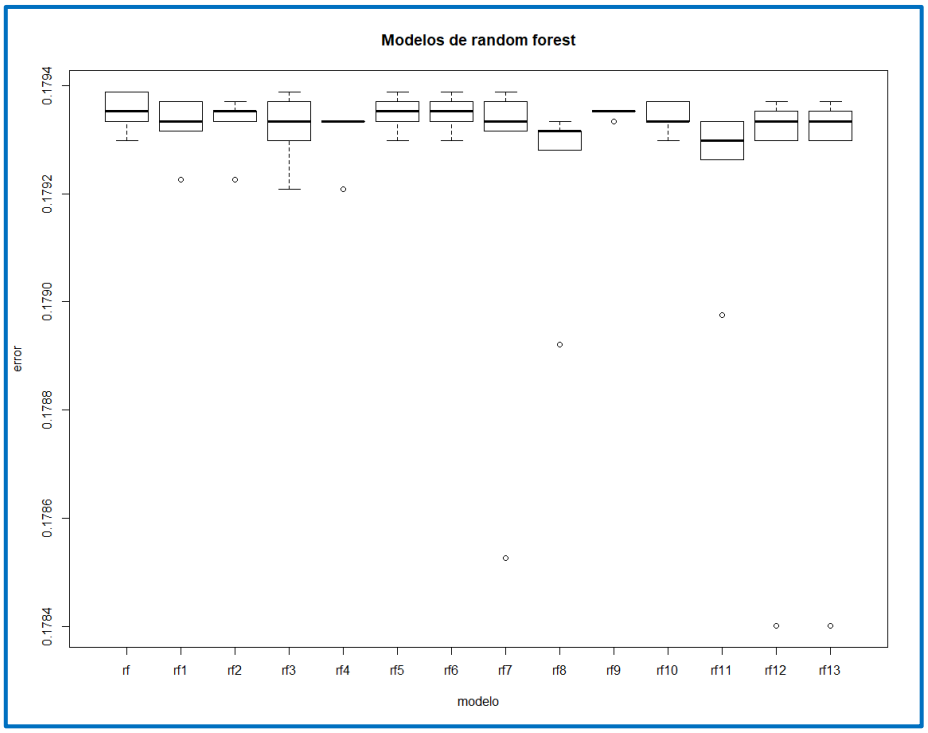


Figura 79: Comparación de modelos de RF

- **Interpretación del modelo óptimo**

  Por último, se procede a calcular la importancia de cada una de las variables independientes. Dicha importancia se establecerá en función de la variabilidad del error cuadrado medio (Figura 80).

  En la Figura 80, podemos apreciar que la variable con mayor importancia en este modelo es "tipo_gasol", la cual corresponde al tipo de gasolina, mientras que el resto de variables aparentemente tiene una importancia bastante inferior a dicha variable.

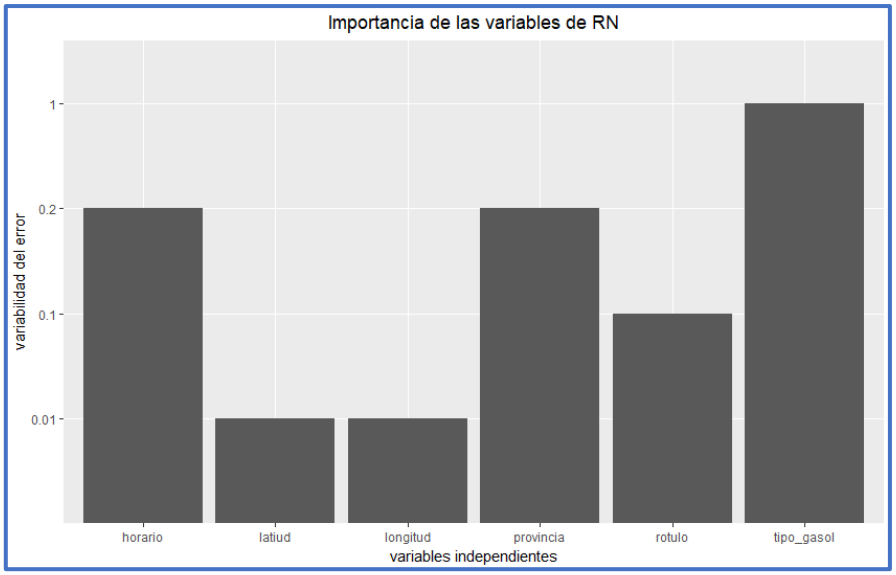


Figura 80: Importancia de las variables independientes del RF óptimo.

10.5 Suport Vector Machine

- **Construcción del modelo**

La construcción del mejor modelo de Suport Vector Machine se lleva a cabo variando los diferentes parámetros propios de esta técnica de forma manual y lógica, como son, el épsilon (margen de error), el valor de C (parámetro de regularización), el número de validaciones cruzadas repetidas, función Kernel (LINEAL, y RADIAL), el valor de gamma (parámetro que requiere el parámetro RADIAL), y la semilla, respectivamente. Esto se llevará a cabo sobre el conjunto de datos de Entrenamiento, para después valorar su funcionamiento sobre la técnica de validación cruzada repetida.

Se han propuesto 20 modelos, de los cuales los 10 primeros utilizan el Kernel LINEAL y, específicamente 5 de ellos son con la semilla 12346, y los otros 5 con otra semilla, 12349; mientras que los 10 siguientes, con el Kernel RADIAL, también los primeros 5 con la semilla 12346, y los 5 restantes con la semilla 12349, para observar si estos modelos presentan el mismo comportamiento a pesar de utilizar una semilla distinta. Dichos modelos se muestran en la Figura 83.

```
#Kernel lineal
set.seed(12346)
svmlineal <- svm(x, y, type="eps-regression", kernel="linear",
                 epsilon=0.01, cost=10, scale=F, cross = 3)
svmlineal1 <- svm(x, y, type="eps-regression", kernel="linear",
                  epsilon=0.02, cost=15, scale=F, cross = 4)
svmlineal2 <- svm(x, y, type="eps-regression", kernel="linear",
                  epsilon=0.01, cost=20, scale=F, cross = 5)
svmlineal3 <- svm(x, y, type="eps-regression", kernel="linear",
                  epsilon=0.02, cost=25, scale=F, cross = 6)
svmlineal4 <- svm(x, y, type="eps-regression", kernel="linear",
                  epsilon=0.01, cost=30, scale=F, cross = 3)
set.seed(12349)
svmlineal5 <- svm(x, y, type="eps-regression", kernel="linear",
                  epsilon=0.01, cost=10, scale=F, cross = 3) #distinta semilla
svmlineal6 <- svm(x, y, type="eps-regression", kernel="linear",
                  epsilon=0.02, cost=15, scale=F, cross = 3)
svmlineal7 <- svm(x, y, type="eps-regression", kernel="linear",
                  epsilon=0.01, cost=20, scale=F, cross = 3)
svmlineal8 <- svm(x, y, type="eps-regression", kernel="linear",
                  epsilon=0.02, cost=25, scale=F, cross = 3)
svmlineal9 <- svm(x, y, type="eps-regression", kernel="linear",
                  epsilon=0.01, cost=30, scale=F, cross = 3)

#Kernel no lineal
set.seed(12346)
svmradial6 <- svm(x,y, type="eps-regression", kernel="radial",
                  epsilon=0.01, gamma=40, cost=5, scale=F, cross = 3)
svmradial7 <- svm(x,y, type="eps-regression", kernel="radial",
                  epsilon=0.02, gamma=40, cost=10, scale=F, cross = 3)
svmradial8 <- svm(x,y, type="eps-regression", kernel="radial",
                  epsilon=0.01, gamma=40, cost=15, scale=F, cross = 3)
svmradial9 <- svm(x,y, type="eps-regression", kernel="radial",
                  epsilon=0.02, gamma=40, cost=20, scale=F, cross = 3)
svmradial10 <- svm(x,y, type="eps-regression", kernel="radial",
                   epsilon=0.01, gamma=40, cost=30, scale=F, cross = 2)
set.seed(12349)
svmradial11 <- svm(x,y, type="eps-regression", kernel="radial",
                   epsilon=0.01, gamma=40, cost=5, scale=F, cross = 3) #distinta semilla
svmradial12 <- svm(x,y, type="eps-regression", kernel="radial",
                   epsilon=0.02, gamma=40, cost=10, scale=F, cross = 4)
svmradial13 <- svm(x,y, type="eps-regression", kernel="radial",
                   epsilon=0.01, gamma=40, cost=15, scale=F, cross = 5)
svmradial14 <- svm(x,y, type="eps-regression", kernel="radial",
                   epsilon=0.02, gamma=40, cost=20, scale=F, cross = 6)
svmradial15 <- svm(x,y, type="eps-regression", kernel="radial",
                   epsilon=0.01, gamma=40, cost=30, scale=F, cross = 2)
```

Figura 83: Modelos propuestos de SVM

- **Comparación de modelos**

  Una vez propuestos los modelos anteriores, vamos a proceder a la comparación de estos. Para ello, vamos a obtener un gráfico de caja en el que se muestra cada uno de los modelos frente a su error medio de predicción en validación cruzada repetida, en el que podemos observar en la Figura 84 que, para la semilla 12346, los modelos tienen el mismo comportamiento que con la semilla 12349 (Figura 85), lo que era de esperar. En este caso, todos los modelos son igual de óptimos ya que con todos ellos se consigue un error en torno a 0.05, 0.1, aproximadamente. Por lo que concluimos que el modelo óptimo de suport vector machine es el primero, *svm*, ya que es el que consigue un menor error de todos.

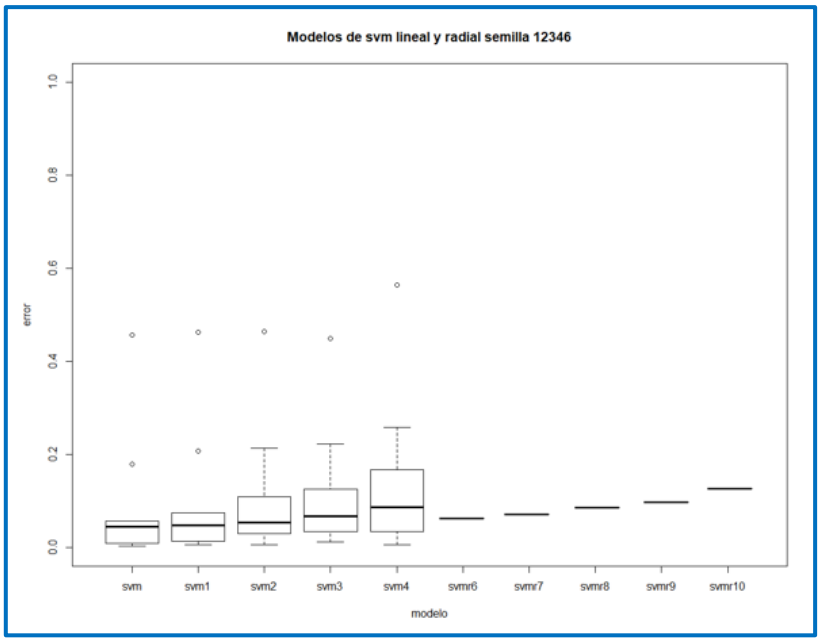


Figura 84: Comparación de modelos de SVM semilla 12346

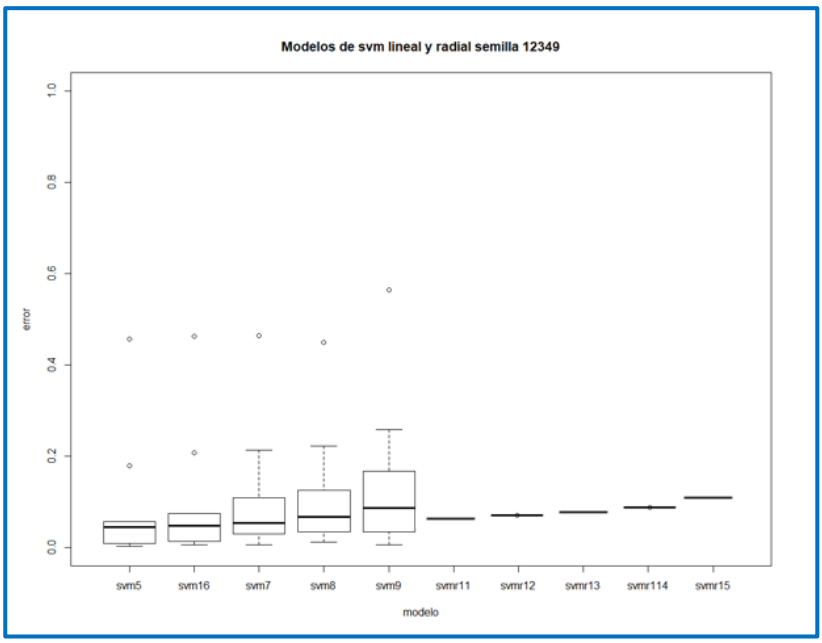


Figura 85: Comparación de modelos de SVM semilla 12349

Por lo que los parámetros del modelo óptimo de Suport Vector Machine *svm*, son:

- Variables a utilizar: las del mejor modelo de regresión lineal (LATITUD, LONGITUD, HORARIO, ROTULO, PROVINCIA, TIPO_GASOLINA).
- Épsilon (margen de error): 0.01
- Valor de C (parámetro de regularización): 10
- Número de validaciones cruzadas repetidas (cross): 3
- Kernel: LINEAL

En el Anexo H, podemos observar la correspondiente salida del gráfico de comparación de modelos de SVM de las Figuras 84-85 más ampliado y separados por comportamientos similares, ya que hay modelos que quizás no se aprecian correctamente.

11 Elección del modelo óptimo

Una vez obtenidos las estructuras más óptimas de los modelos propuestos anteriormente, vamos a determinar cuál de todos estos es el mejor modelo de predicción, es decir, cuál es la técnica estadística que mejor predice el precio del carburante de las gasolineras de España.

Para ello, nos fijaremos, en el error de la raíz cuadrática media o error de predicción calculado sobre la técnica de validación cruzada repetida para cada uno de estos modelos.

En la Tabla 4, se muestra un resumen de los mismos, en la que se puede observar que el modelo que minimiza el error de predicción es el de Suport Vector Machine. Dicho modelo tiene una diferencia significativa con respecto al resto de modelos y más especialmente si lo comparamos con el modelo de Regresión Lineal, el cual, es la única alternativa que ofrece la posibilidad de ser interpretado.

Modelo	Error de predicción
Regresión Lineal	0.1791818
Redes Neuronales	0.1510482
Random Forest	0.1793235
Suport Vector Macchine	0.0860987

Tabla 5: Resumen comparativo de los modelos óptimos

12 Análisis de la competencia empresarial mediante teoría de juegos

Una vez obtenido el modelo que mejor predice el precio del carburante de las gasolineras de España utilizando distintas técnicas de Machine Learning, vamos a ofrecer otra forma de predecir el precio de la gasolina, en este caso, nos vamos a basar en teoría de juegos, más concretamente en el Juego de Bertrand. Es decir, vamos a conseguir predecir cuál sería la estrategia más óptima para cada una de las gasolineras de una zona de España teniendo en cuenta su entorno. Con estrategia más óptima, nos referimos al concepto de Equilibrio de Nash.

Hemos decidido escoger una zona con pocas gasolineras ya que así se trataría de un claro ejemplo de competencia, aunque se puede replicar para cualquier ciudad de la misma forma.

En este caso, vamos a elegir La Gomera, ya que como bien mencionamos anteriormente es un claro ejemplo de competencia ya que, al ser una isla, sus habitantes no pueden ir a echar gasolina a las otras islas.

A continuación, vamos a situar en un mapa las gasolineras que se encuentran en La Gomera, donde podemos observar que hay 4 (puntos de color negro) (Figura 90). Por lo que vamos a realizar una posible disposición de las gasolineras en ese mapa, la cual se muestra en la Figura 91 (en este caso, hemos realizado esta forma, pero podría ser cualquier otra).

Por tanto, este juego se caracteriza por:

- Número de jugadores/gasolineras: 4
- Estrategias: $p_1 \in (0, \infty)$, $p_2 \in (0, \infty)$, $p_3 \in (0, \infty)$, y $p_4 \in (0, \infty)$
- Función de beneficio de cada una de las empresas, ingresos menos costes:

  $\pi_1(p_1, p_2, p_3, p_4) = p_1 * q_1(p_1, p_2, p_3, p_4) - c*q_1(p_1, p_2, p_3, p_4) = (p_1 - c) * q_1(p_1, p_2, p_3, p_4) - c_f$
  $\pi_2(p_1, p_2, p_3, p_4) = p_2 * q_2(p_1, p_{2,,}p_3, p_4) - c*q_2(p_1, p_2, p_3, p_4) = (p_2 - c) * q_2(p_1, p_2, p_3, p_4) - c_f$
  $\pi_3(p_1, p_2, p_3, p_4) = p_3 * q_3(p_1, p_2, p_3, p_4) - c*q_3(p_1, p_2, p_3, p_4) = (p_3 - c) * q_3(p_1, p_2, p_3, p_4) - c_f$
  $\pi_4(p_1, p_2, p_3, p_4) = p_4 * q_4(p_1, p_2, p_3, p_4) - c*q_4(p_1, p_2, p_3, p_4) = (p_4 - c) * q_4(p_1, p_2, p_3, p_4) - c_f$

donde:

- p_1, p_2, p_3, y p_4 son los precios de la gasolinera 1, 2, 3, y 4, respectivamente
- $q_1(p_1, p_2, p_3, p_4)$, $q_2(p_1, p_{2,,} p_3, p_4)$, $q_3(p_1, p_{2,,} p_3, p_4)$, y $q_4(p_1, p_{2,,} p_3, p_4)$ son las funciones de demanda de la gasolinera 1, 2, 3, y 4, respectivamente.
- c es el coste
- c_f es el coste fijo

Por tanto, necesitamos conocer la demanda de cada una de las gasolineras para así obtener su beneficio. En este caso, debido a que en nuestra base de datos no disponemos de una variable que se refiera a la demanda, y por falta de tiempo hemos partido de una función de demanda que nos parece lógica teniendo en cuenta la disposición de las gasolineras (Figura 91). Pero en la realidad se debería de hacer un estudio histórico de los datos, es decir, del precio y lo que vendió cada gasolinera. Esto último, se podría realizar llevando a cabo un muestreo, es decir, una persona se situaría delante de cada gasolinera y anotaría el número de coches que irían, y el número de personas que echarían diésel o gasolina. Con todo ello, se realizaría un modelo con los datos obtenidos.

La función de demanda para la gasolinera 1 es:

$$q_1(p_1, p_2, p_3, p_4) = \frac{Q}{4}(a - p_1) - b_2^1 \alpha (p_1 - p_2) - b_3^1 \alpha (p_1 - p_3) - b_4^1 \alpha (p_1 - p_4)$$

De forma simétrica, las funciones de demanda para la gasolinera 2, 3, y 4, son:

$$q_2(p_1, p_2, p_3, p_4) = \frac{Q}{4}(a - p_2) - b_1^2 \alpha (p_2 - p_1) - b_3^2 \alpha (p_2 - p_3) - b_4^2 \alpha (p_2 - p_4)$$

$$q_3(p_1, p_2, p_3, p_4) = \frac{Q}{4}(a - p_3) - b_1^3 \alpha (p_3 - p_1) - b_2^3 \alpha (p_3 - p_2) - b_4^3 \alpha (p_3 - p_4)$$

$$q_4(p_1, p_2, p_3, p_4) = \frac{Q}{4}(a - p_4) - b_1^4 \alpha (p_4 - p_1) - b_2^4 \alpha (p_4 - p_2) - b_3^4 \alpha (p_4 - p_3)$$

donde:

- Q es la consumo final de esta isla
- a es el precio máximo de la gasolina, siendo $a > c$
- b_2^1 es el parámetro geográfico de la gasolinera 1 teniendo en cuenta la gasolinera 2. Es decir, cuanto más cercana esté la gasolinera 2 de la 1, este valor será mayor, mientras que cuanto esté menos cerca, menor será.

- α es el parámetro que mide la flexibilidad que tiene una persona de moverse a la competencia.

Asumimos que D = 2000, c_f = 500, c = 0.6, y a = 1.8.

Para determinar los valores de $b^{gasolinera}_{gasolinera\_competencia}$, los vamos a fijar en función de lo cercano o no que esté la competencia. Es decir, si la gasolinera está cerca de la competencia (competencia directa), el valor de b será alto; mientras que si la gasolinera está lejos (competencia lejana), el valor de b será bajo. La cercanía o la lejanía de las gasolineras lo vamos a observar en la Figura 91. En este caso, la competencia lejana será $b = 1000$, 100 para la competencia media, y 70 para la competencia lejana. De forma que los valores de $b^{gasolinera}_{gasolinera\_competencia}$ son:

$$\begin{array}{lll} b_2^1 = 100, & b_3^1 = 70, & b_4^1 = 70 \\ b_1^2 = 100, & b_3^2 = 100, & b_4^2 = 100 \\ b_1^3 = 70, & b_2^3 = 100, & b_4^3 = 1000 \\ b_1^4 = 70, & b_2^4 = 100, & b_3^4 = 1000 \end{array}$$

Asumiendo todos los valores anteriores, las funciones de demanda y de beneficios quedarían,

$$q_1(p_1, p_2, p_3, p_4) = 500\,(1.8 - p_1) - 100\alpha\,(p_1 - p_2) - 70\alpha\,(p_1 - p_3) - 70\alpha\,(p_1 - p_4)$$

$$q_2(p_1, p_2, p_3, p_4) = 500\,(1.8 - p_2) - 100\alpha\,(p_2 - p_1) - 100\alpha\,(p_2 - p_3) - 100\alpha\,(p_2 - p_4)$$

$$q_3(p_1, p_2, p_3, p_4) = 500\,(a - p_3) - 70\alpha\,(p_3 - p_1) - 100\alpha\,(p_3 - p_2) - 1000\alpha\,(p_3 - p_4)$$

$$q_4(p_1, p_2, p_3, p_4) = 500\,(a - p_4) - 70\alpha\,(p_4 - p_1) - 100\alpha\,(p_4 - p_2) - 1000\alpha\,(p_4 - p_3)$$

$$\pi_1(p_1, p_2, p_3, p_4) = (p_1 - 0.6) * q_1(p_1, p_2, p_3, p_4) - 500$$
$$\pi_2(p_1, p_2, p_3, p_4) = (p_2 - c) * q_2(p_1, p_2, p_3, p_4) - 500$$
$$\pi_3(p_1, p_2, p_3, p_4) = (p_3 - c) * q_3(p_1, p_2, p_3, p_4) - 500$$
$$\pi_4(p_1, p_2, p_3, p_4) = (p_4 - c) * q_4(p_1, p_2, p_3, p_4) - 500$$

Por tanto, una vez que ya hemos determinado todos los parámetros, vamos a predecir los precios de cada una de las gasolineras en función de α, paramétro de flexibilidad, para conocer cómo varían los precios de las gasolineras sabiendo lo flexible que es una persona de moverse a la competencia.

- *Vamos a realizar el caso cuando no hay competencia, es decir, con* $\alpha = 0$.

  La función de beneficios de cada una de las gasolineras queda de la siguiente forma:

  $\pi_1(p_1, p_2, p_3, p_4) = ((p_1 - 0.6) * (500 * (1.8 - p_1))) - 500$
  $\pi_2(p_1, p_2, p_3, p_4) = ((p_2 - 0.6) * (500 * (1.8 - p_2))) - 500$
  $\pi_3(p_1, p_2, p_3, p_4) = (p_3 - 0.6) * (500 * (1.8 - p_3))) - 500$
  $\pi_4(p_1, p_2, p_3, p_4) = (p_4 - c) * (500 * (1.8 - p_4))) - 500$

  Obtenemos las funciones de mejor respuesta de la gasolinera 1, 2, 3, y 4, de forma separada. Para ello se busca el máximo de la función de beneficios de una gasolinera fijando las otras. Es decir,

  La función de mejor respuesta de la gasolinera 1 (G_1) es:

  $$\frac{d\,\pi_1(p_1, p_2, p_3, p_4)}{dp_1} = \frac{d\,(((p_1 - 0.6) * (500 * (1.8 - p_1))) - 500)}{dp_1} = 0$$

  $$\Rightarrow \quad p_1 = 1.2$$

  La función de mejor respuesta de la gasolinera 2 (G_2) es:

  $$\frac{d\,\pi_2(p_1, p_2, p_3, p_4)}{dp_2} = \frac{d\,(((p_2 - 0.6) * (500 * (1.8 - p_2))) - 500)}{dp_2} = 0$$

  $$\Rightarrow \quad p_2 = 1.2$$

  La función de mejor respuesta de la gasolinera 3 (G_3) es:

  $$\frac{d\,\pi_3(p_1, p_2, p_3, p_4)}{dp_3} = \frac{d\,(((p_3 - 0.6) * (500 * (1.8 - p_3))) - 500)}{dp_3} = 0$$

  $$\Rightarrow \quad p_3 = 1.2$$

  La función de mejor respuesta de la gasolinera 4 (G_4) es:

  $$\frac{d\,\pi_4(p_1, p_2, p_3, p_4)}{dp_4} = \frac{d\,(((p_4 - 0.6) * (500 * (1.8 - p_4))) - 500)}{dp_4} = 0$$

  $$\Rightarrow \quad p_4 = 1.2$$

  Por tanto, el Equilibrio de Nash (EN) sería $(p_1, p_2, p_3, p_4) = (1.2, 1.2, 1.2, 1.2)$.

  En el caso de no haber competencia, el precio de la gasolina para todas las gasolineras sería 1.2, que corresponde con el precio máximo de las islas Canarias,

aproximadamente (cabe recordar que como mencionamos anteriormente, el precio máximo de las islas Canarias es 1.15).

Para los demás casos de α, se realiza de la misma forma.

- *Para el caso de haber una flexibilidad de 0.5 ($\alpha = 0.5$),* obtenemos que el EN sería $(p_1, p_2, p_3, p_4) = (1.13, 1.11, 0.99, 0.99)$.

- *Para el caso de no haber flexibilidad ($\alpha = 1$),* obtenemos que el EN sería $(p_1, p_2, p_3, p_4) = (1.05, 1.04, 0.88, 0.88)$.

- *Para el caso de haber una flexibilidad de 1.5 ($\alpha = 1.5$),* obtenemos que el EN sería $(p_1, p_2, p_3, p_4) = (1.01, 0.98, 0.83, 0.83)$.

- *Para el caso de haber una flexibilidad de 2 ($\alpha = 2$),* obtenemos que el EN sería $(p_1, p_2, p_3, p_4) = (0.97, 0.94, 0.8, 0.8)$.

- *Para el caso de haber una flexibilidad de 3 ($\alpha = 3$),* obtenemos que el EN sería $(p_1, p_2, p_3, p_4) = (0.85, 0.82, 0.78, 0.78)$.

Podemos observar que a medida que aumentamos el valor de la flexibilidad, los precios de las gasolineras van disminuyendo, es decir, el hecho de haber competencia hace que los precios sean más bajos. Además, el precio de las gasolineras 1 y 2, y 3, y 4 es similar, y dispares entre las demás, lo cual también se puede ver en la Figura 92. Esto es debido a que entre las últimas gasolineras hay una competencia cercana, y entre las demás hay una competencia lejana, por lo que poseen precios similares (Figura 91). Por tanto, podemos afirmar que el precio de la gasolina está impuesto atendiendo a las demás gasolineras (cuánto más próximas estén, los precios serán más parecidos, y viceversa).

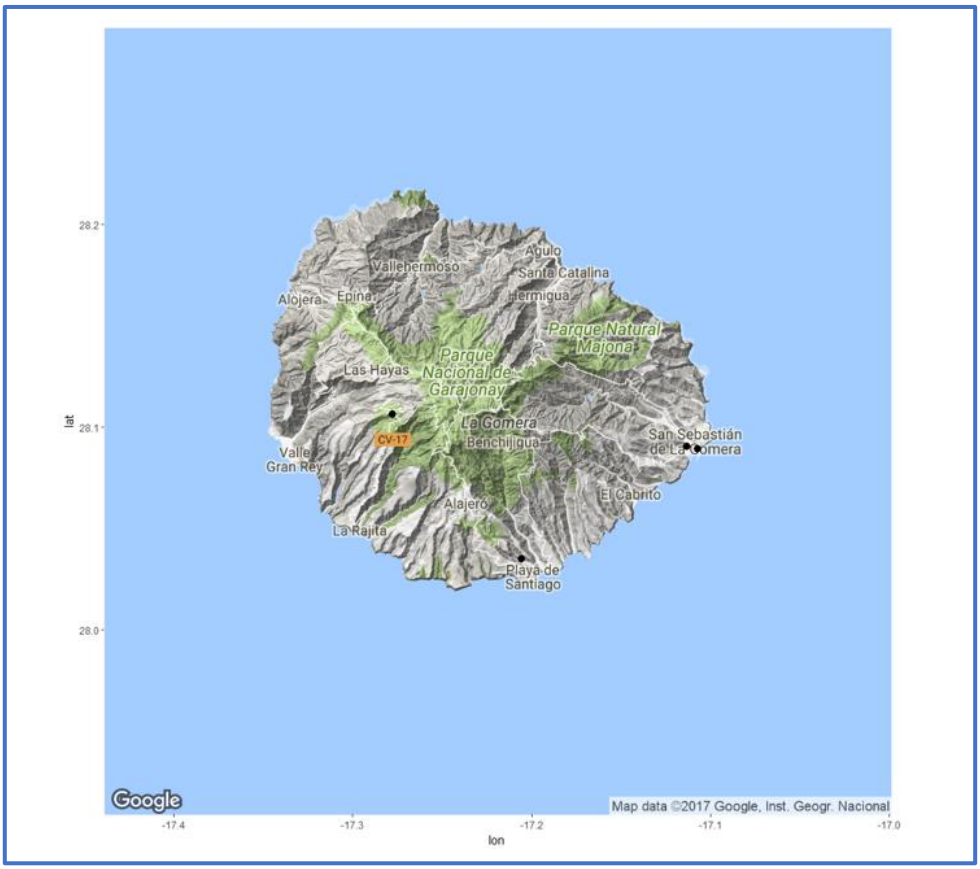


Figura 90: Gasolineras de La Gomera

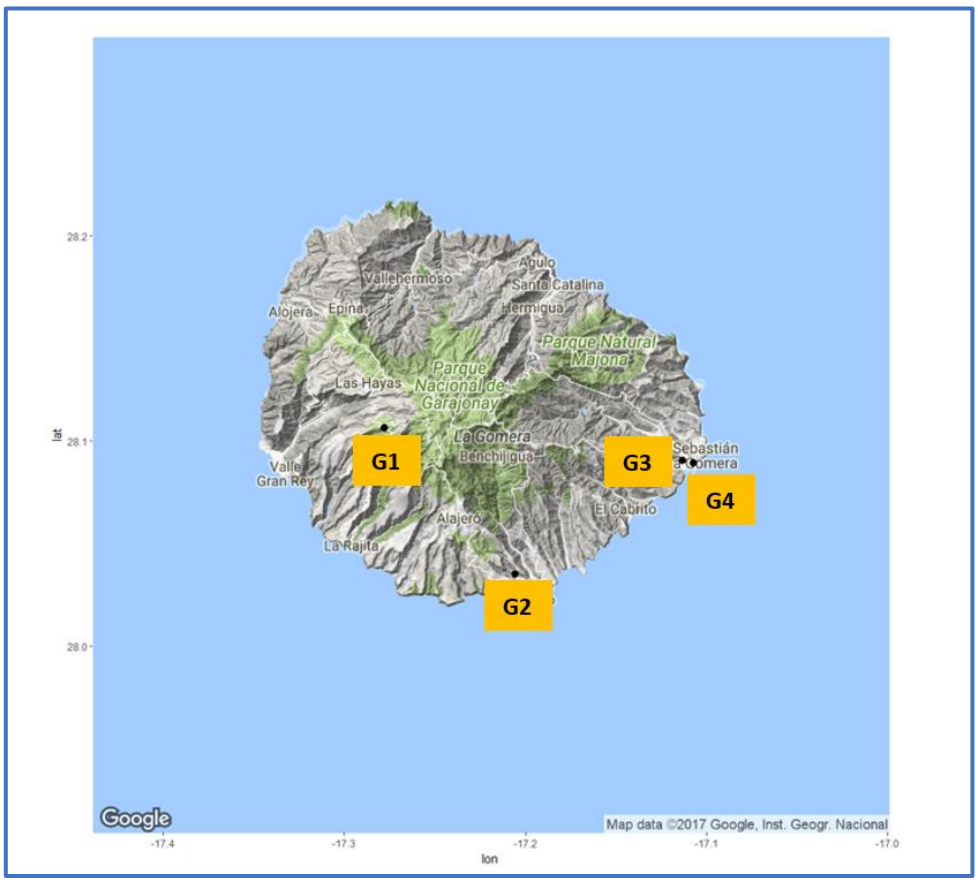


Figura 91: Disposición de las gasolineras de La Gomera

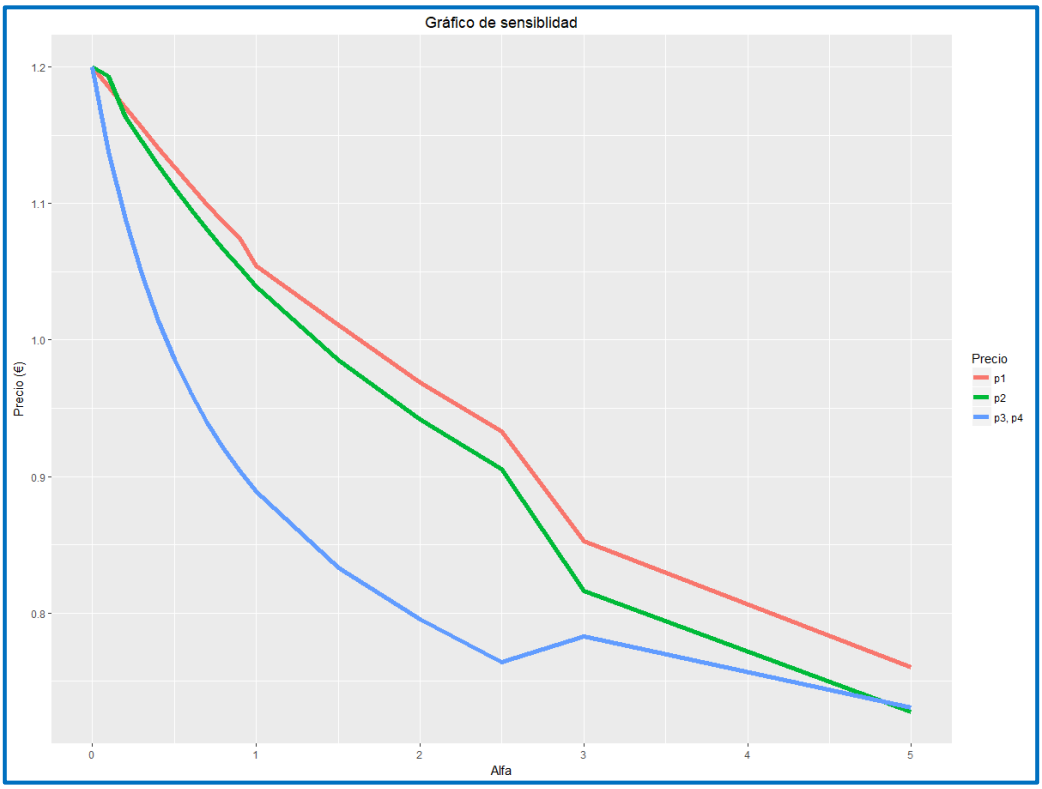


Figura 92: Gráfico de sensibilidad

13 Conclusiones

Durante la memoria, hemos conocido en profundidad el precio de las gasolineras de España, estableciendo posibles relaciones. Todo esto se realizó por medio de técnicas multivariantes y descriptivos.

Por consiguiente, hemos buscado comprender la relación del precio de los distintos establecimientos con respecto a las distintas características físicas de cada gasolinera, cuyo fin era entender cómo esas características afectaban a la gasolinera a la hora de establecer un precio. Esto se ha realizado mediante técnicas de Machine Learning, y mediante teoría de juegos analizando la competencia ya que un establecimiento marca sus tarifas atendiendo no sólo a sus características individuales, sino también a las del resto de las gasolineras.
En el primer caso, hemos obtenido un mejor modelo de predicción, el de SVM con un error de predicción, de 0.08, aproximadamente, en el que el precio está establecido en función de la latitud, la longitud, la provincia, el tipo de gasolina, el horario, y el rótulo. El único inconveniente es que este modelo carece de interpretación.
En el segundo, hemos obtenido cuál sería el precio óptimo de cada gasolinera, teniendo en cuenta la competencia. Para simplificar, lo realizamos para La Gomera, pero de igual forma se podría hacer para cualquier otra zona. En el que concluimos que si no hay competencia todas las gasolineras de La Gomera marcan el precio más alto

$(p_1, p_2, p_3, p_4) = (1.2, 1.2, 1.2, 1.2)$, mientras que si la hay, el precio es más bajo atendiendo a las tarifas que marca la competencia más cercana o lejana (si la competencia es cercana, los precios serán parecidos, mientras que si la competencia es lejana, precios dispares).

13.1 Mejoras y fututo trabajo

A continuación, se detallan ciertos aspectos que podrían suponer una continuación de este estudio y que, lamentablemente no se han podido desarrollar por razones de tiempo y coste computacional.

- Realización de algún otro contraste, para observar más relaciones. Es decir, observar si en función de las comunidades autónomas el precio es más caro o no, para comprobar la homogeneidad de impuestos.

- Análisis en profundidad del estudio predictivo, utilizando una búsqueda en rejilla de valores para la parametrización de los distintos parámetros de las técnicas de Machine Learning. Además, del uso de otra técnica, como es, Gradient Boosting, que es otro algoritmo típico relativo a árboles de decisión.

- Análisis más profundo de la competencia en las gasolineras. Es decir, cerrar una gasolinera de La Gomera y ver cómo va variando el precio de las otras gasolineras. Luego cerrar otra y volver a predecir los precios, y así hasta que todas las gasolineras estén cerradas para observar cómo van cambiando los precios cuando la competencia es menor.

Bibliografía

Bibliografía referenciada en el texto

[1] Sitio web. http://www.laregion.es/articulo/historia-en-4-tiempos/primeras-gasolineras/20100422104543462281.html. 2017

[2] Sitio web. http://www.rtve.es/noticias/20150603/numero-gasolineras-crece-203-espana-desde-inicio-crisis-2007/1156421.shtml. 2017

[3] Análisis Multivariante. Quinta edición. Hair, Anderson. Tatham. Black. Editorial Prentice Hall. 1999

[4] Cluster Analysis. En Data Analysis, Classification and Related MEthods.Belgium: Springer. Kiers, H., & Rasson, J.-P. 2000

[5] Análisis de Datos Multivariantes. Daniel Peña. S.A. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA. 2002

[6] Análisis Multivariante I, J.L.Valencia Delfa, y M.L. Vicente Hernanz. Editorial CERSA. 2014

[7] Análisis multivariable para las ciencias sociales. Jean-Pierre Lévy Mangin y Jesús Varela Mallou. Editorial PEARSON EDUCATION, S.A, PRENTICE HALL. 2003

[8] Introducción al análisis de regresión lineal. Montgomery-Peck-Vining. 3ª edición. Editorial CECSA. 2002

[9] Aplicación de las redes neuronales artificiales a la regresión. Quitín Martín Martín y Yanira del Rosario De Paz Santana. Editorial La Muralla, S.A. 2007

[10] Stephen Marsland. Machine Learning: An Algorithmic Perspective. Chapman & Hall/CRC, 1st edition, 2009

[11] Pattern Recognition and Machine Learning. Christopher M. Bishop. Editorial Springer. 2006

[12] Random Forest. Machine Learning. Breiman, L. 2001

[13] *Data Mining with Decision Trees: Theory and Applications*. World Scientific Publishing Co., Inc. River Edge. Lior Rokach and Oded Maimon. 2008.

[14] Sitio web. http://www.saedsayad.com/support_vector_machine_reg.htm. 2017

[15] Sitio web. https://es.wikipedia.org/wiki/Validaci%C3%B3n_cruzada. 2017

[16] Sitio web. http://www.sgapeio.es/INFORMEST/VICongreso/taller/applets/biomates/explora/explora_grubbs/explora_grubbs.htm. 2017

[17] Un primer curso de teoría de juegos (Robert Gibbons, Traducido por: Calvo, y Vila). 1992

[18] Material Didáctico. Asignatura "Competencia empresarial y teoría de juegos". Javier Castro. Máster en Minería de Datos e Inteligencia de Negocios. Madrid. 2016

[19] Material Didáctico. Asignatura "Técnicas y Metodología de la Minería de Datos". Aida Calviño. Máster en Minería de Datos e Inteligencia de Negocios. Madrid. 2016

[20] Sitio web. http://lasoga.org/una-mente-maravillosa-el-equilibrio-de-nash-y-el-asesinato-de-kitty-genovese/. 2017

[21] Sitio web. https://es.wikipedia.org/wiki/Equilibrio_de_Nash. 2017

[22] Sitio web. http://www.consumer.es/web/es/motor/mantenimiento_automovil/2007/08/28/166039.php. 2017

[23] Sitio web. http://www.minetad.gob.es/energia/petroleo/faq/Paginas/Faqs.aspx. 2017

[24] Sitio web. http://www.gasoilalmejorprecio.com/tipos-de-gasoleo-diferencias-entre-el-gasoleo-a/. 2017

[25] Sitio web. https://www.elespanol.com/ciencia/20160630/136486733_0.html. 2017

[26] Sitio web. https://noticias.coches.com/consejos/diferencias-entre-glp-y-gnc-que-combustible-es-mejor/86314. 2017

[27] Sitio web. https://es.wikipedia.org/wiki/Gas_natural_comprimido. 2017

[28] Sitio web. https://es.wikipedia.org/wiki/Biodi%C3%A9sel . 2017

[29] Sitio web. http://www.ub.edu/ecologia/carlos.gracia/PublicacionesPDF/Cap%C3%ADtulo%204_Bioetanol.pdf. 2017

[30] Sitio web. https://es.wikipedia.org/wiki/F%C3%B3rmula_del_haversine. 2017

[31] Sitio web. https://en.wikipedia.org/wiki/Multimodal_distribution. 2017

[32] Fundamentos de estadística. Daniel Peña. Alianza Editorial. 2001.

[33] Sitio web. https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico. 2017

[34] Sitio web. https://www.investopedia.com/terms/g/gametheory.asp. 2017

Bibliografía no referenciada en el texto

INTRODUCCIÓN

Sitio web. http://www.zonaeconomica.com/teoria-de-juegos. 2017

IMPORTACIÓN

Sitio web. https://cran.r-project.org/web/packages/anytime/anytime.pdf. 2017
Sitio web. https://stackoverflow.com/questions/25655576/convert-rfc-3339-timestamp-r-language. 2017
Sitio web. https://stackoverflow.com/questions/22009276/trouble-with-date-format-using-the-function-as-posixct-in-r. 2017

ANÁLISIS DESCRIPTIVO

Sitio web. http://r4stats.com/examples/graphics-ggplot2/.2017
Sitio web. https://stat.ethz.ch/R-manual/R-devel/library/base/html/substr.html. 2017
Sitio web. https://www.stat.berkeley.edu/classes/s133/saving.html. 2017

VARIABLES DE LA COMPETENCIA

Sitio web. https://stackoverflow.com/questions/32363998/function-to-calculate-geospatial-distance-between-two-points-lat-long-using-r. 2017
Sitio web. https://www.r-bloggers.com/how-to-write-the-first-for-loop-in-r/.2017
Sitio web. https://cran.r-project.org/web/packages/geosphere/geosphere.pdf. 2017
Sitio web. https://es.wikipedia.org/wiki/F%C3%B3rmula_del_haversine. 2017
Sitio web. https://stackoverflow.com/questions/21585721/how-to-create-an-empty-matrix-in-r. 2017

RECODIFICACIÓN DE LAS CATEGORÍAS DE ALGUNA DE LAS VARIABLES

Sitio web. https://cran.r-project.org/doc/contrib/Chicana-Introduccion_al_uso_de_R.pdf. 2017

VALORES ATÍPICOS

Sitio web. https://cran.r-project.org/web/packages/outliers/outliers.pdf. 2017
Sitio web. https://rpro.wikispaces.com/Valores+at%C3%ADpicos+%28outliers%29?responseToken=7580faa1b0a472fe31ed90d577b1e9c5. 2017
Sitio web. https://rpro.wikispaces.com/Valores+at%C3%ADpicos+%28outliers%29. 2017
Sitio web. https://stat.ethz.ch/pipermail/r-help/2005-April/069471.html. 2017
Sitio web. http://www.ennaranja.com/economia-facil/por-que-suben-o-bajan-las-empresas-el-precio-de-sus-productos/.2017

ANÁLISIS MULTIVARIANTE

Contraste

Sitio web. http://www.scientific-european-federation-osteopaths.org/los-tests-estadisticos/.2017
Sitio web. http://rchibchombia.blogspot.com.es/2011/01/prueba-de-u-mann-whitney-para-muestras.html. 2017
Sitio web. http://www.r-tutor.com/elementary-statistics/non-parametric-methods/mann-whitney-wilcoxon-test. 2017

Análisis clúster

Sitio web. https://www.r-bloggers.com/k-means-clustering-in-r/.2017
Sitio web. https://dlegorreta.wordpress.com/2015/03/18/analisis-de-cluster-un-ejemplo-sencillo/.2017
Sitio web. http://analisisydecision.es/manual-curso-introduccion-de-r-capitulo-15-analisis-cluster-con-r-ii/178/.2017
Sitio web. https://rstudio-pubs-static.s3.amazonaws.com/33876_1d7794d9a86647ca90c4f182df93f0e8.html. 2017
Sitio web. http://rstudio-pubs-static.s3.amazonaws.com/97848_cda6939060334a83b0e53b222b1e4b52.html. 2017
Sitio web. http://analisisydecision.es/grafico-de-correlaciones-entre-variables/.2017
Sitio web. http://ggplot2.tidyverse.org/reference/geom_abline.html. 2017
Sitio web. https://rpubs.com/Rortizdu/140201. 2017
Fundamentos de Estadística. Daniel Peña. Ciencias Sociales. Alianza Editorial. 2001
Problemas de probabilidades y estadística. Vol 2: Inferencia estadística. Carles M. Cuadras. 1991

MODELOS PREDICTIVOS

División del conjunto de datos

Sitio web. https://stackoverflow.com/questions/17200114/how-to-split-data-into-training-testing-sets-using-sample-function. 2017

Modelos:

Sitio web. http://analisisydecision.es/muestreo-de-datos-con-r/.2017
Sitio web. https://stackoverflow.com/questions/20782583/how-to-convert-from-category-to-numeric-in-r. 2017
Sitio web. https://www.r-bloggers.com/scalable-machine-learning-for-big-data-using-r-and-h2o/amp/.2017
Sitio web. https://github.com/h2oai/h2o-tutorials/blob/master/tutorials/deeplearning/deeplearning.md. 2017
Sitio web. https://cran.r-project.org/web/packages/e1071/e1071.pdf. 2017

A Anexo descriptivo de las variables de una gasolinera

Variables cuantitativas

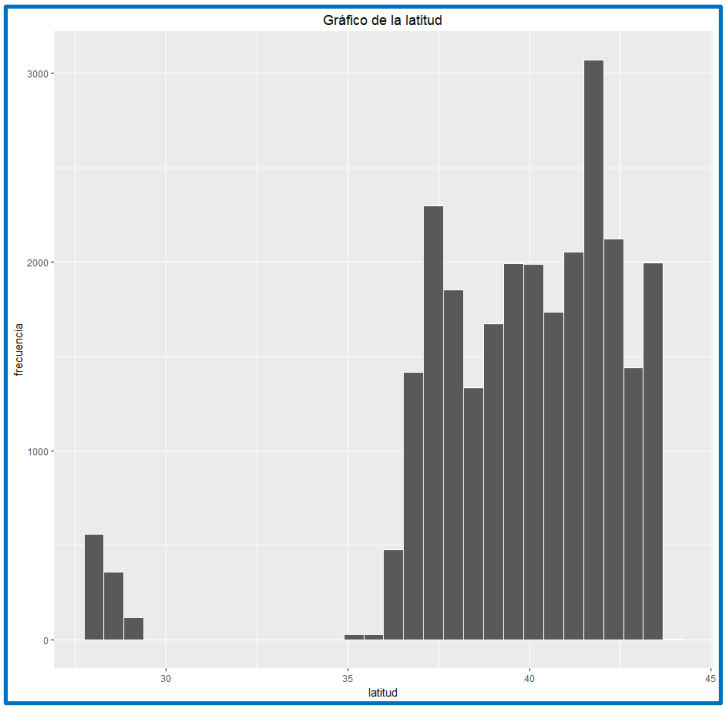


Figura 1: Gráfico de la latitud

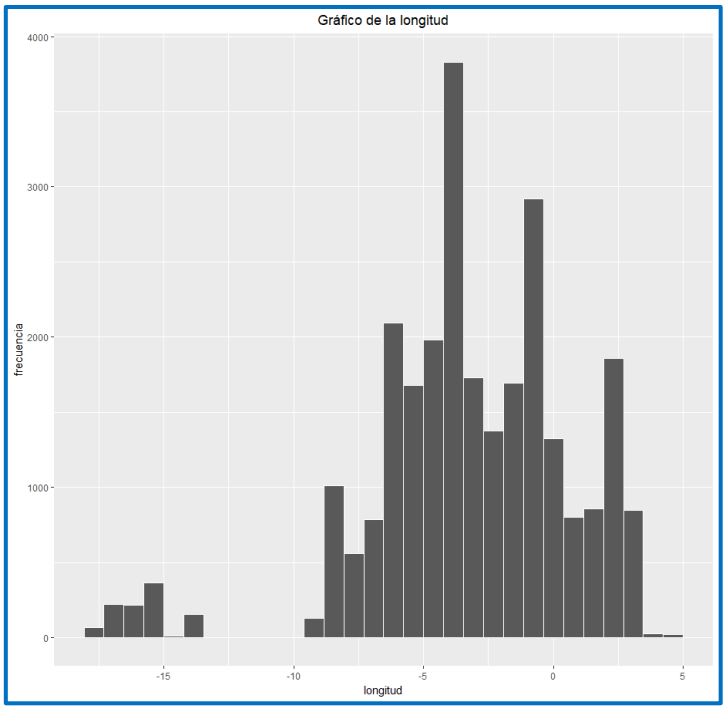


Figura 2: Gráfico de la longitud

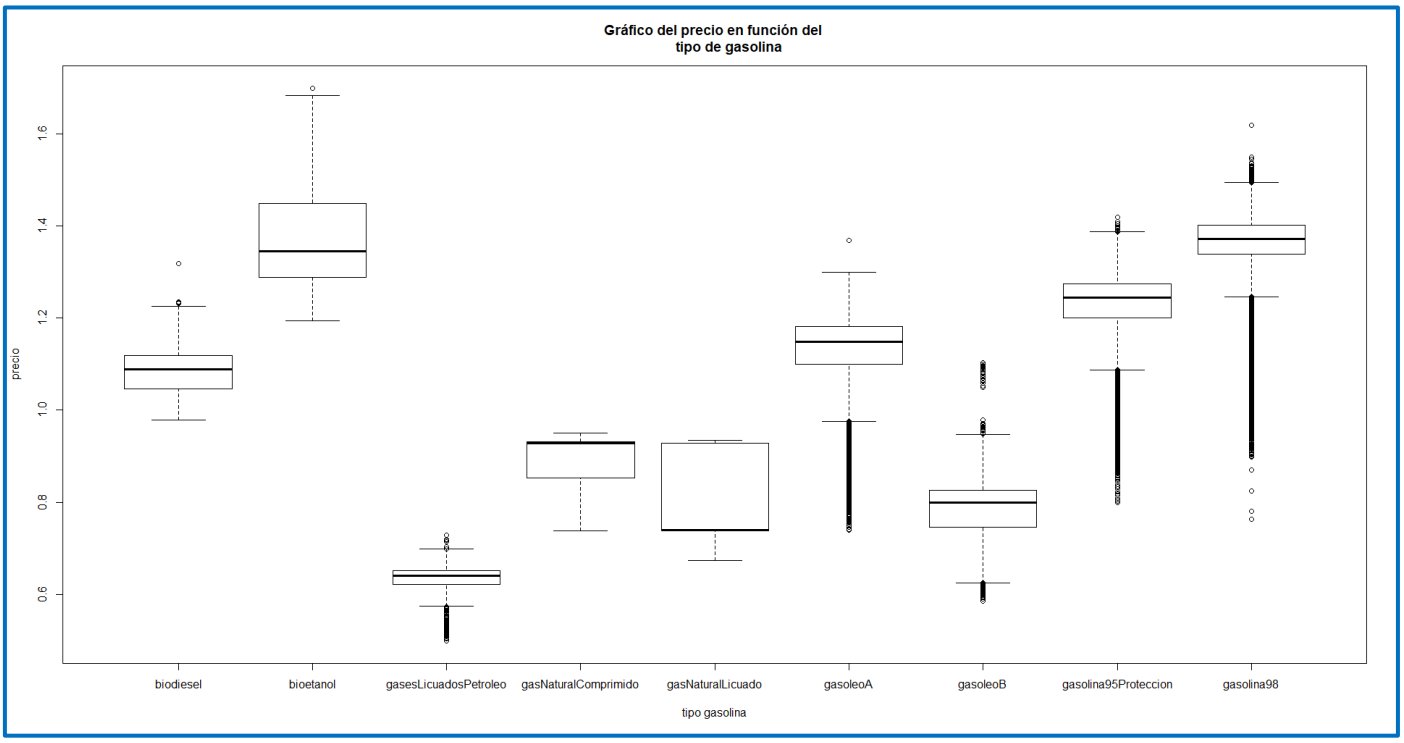


Figura 3: Gráfico del precio en función del tipo de gasolina

Variables cualitativas

```
> CpMasFrecuente = head(sort(table(gasolineras$cp), decreasing=T), 9)
> CpMasFrecuente

04700 30500 41500 29680 30800 29200 29600 35500 46500
   69    63    62    55    55    54    52    50    50
```

Figura 4: Código postales más frecuentes

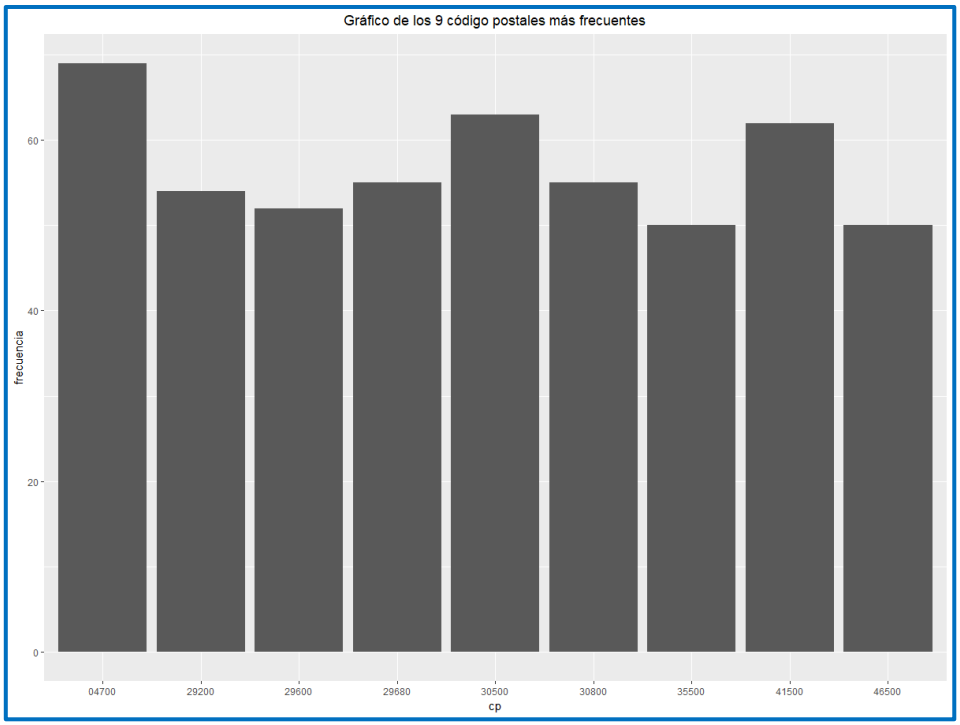


Figura 5: Gráfico de los 9 códigos postales más frecuentes

```
> HorarioMasFrecuente = head(sort(table(gasolineras$horario), decreasing=T), 20)
> HorarioMasFrecuente

       L-D: 24H L-D: 06:00-22:00 L-D: 07:00-23:00 L-D: 07:00-22:00 L-D: 06:00-23:00 L-D: 06:00-00:00 L-D: 06:30-22:30   L: 06:00-22:00   L: 07:00-23:00
          11672             4236             2654              892              794              721              380              371              363
L-D: 07:00-21:00   L: 07:00-22:00 L-D: 06:00-23:59 L-D: 08:00-22:00 L-D: 06:30-22:00   L: 06:00-23:00 L-D: 05:00-23:00 L-S: 06:00-22:00 L-D: 06:30-23:00
               279              242              195              182              128              127              126              117               97
L-S: 07:00-22:00 L-D: 07:00-00:00
                95               92
```

Figura 6: Horarios más frecuentes

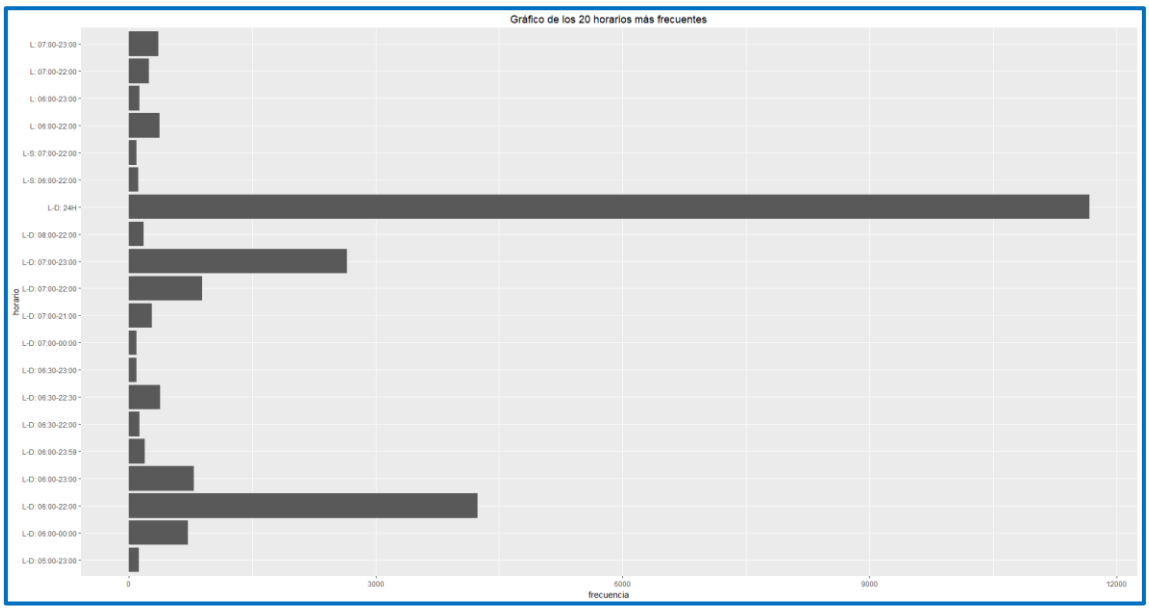


Figura 7: Gráfico de los 20 horarios más frecuentes

```
> LocalidadMasFrecuente = head(sort(table(gasolineras$localidad), decreasing=T), 39)
> LocalidadMasFrecuente

                  MADRID                BARCELONA                  SEVILLA                 VALENCIA                   MALAGA
                     509                      246                      160                      154                      140
               ZARAGOZA                   PALMA                CORDOBA PALMAS DE GRAN CANARIA (LAS)  JEREZ DE LA FRONTERA
                     128                      122                      115                      107                       98
     ALICANTE/ALACANT                VALLADOLID                 TERRASSA     CASTELLON DE LA PLANA                 ALBACETE
                      95                       94                       92                       87                       78
          ELCHE/ELX HOSPITALET DE LLOBREGAT (L')                   MURCIA                    GIJON                  ALCORCON
                      76                       76                       75                       74                       71
              BADAJOZ                    VIGO                  GRANADA  SANTA CRUZ DE TENERIFE                 SANTANDER
                      70                       68                       65                       65                       64
           TARRAGONA                 EJIDO (EL)                 SABADELL                   LLEIDA  ALCALA DE GUADAIRA
                      64                       63                       63                       62                       60
  ALCALA DE HENARES           MOLINA DE SEGURA            VITORIA-GASTEIZ                     REUS                CARTAGENA
                      60                       60                       59                       57                       55
            MARBELLA                  ANTEQUERA                CORUÑA (A)                TORREVIEJA
                      55                       54                       54                       54
```

Figura 8: Localidades más frecuentes

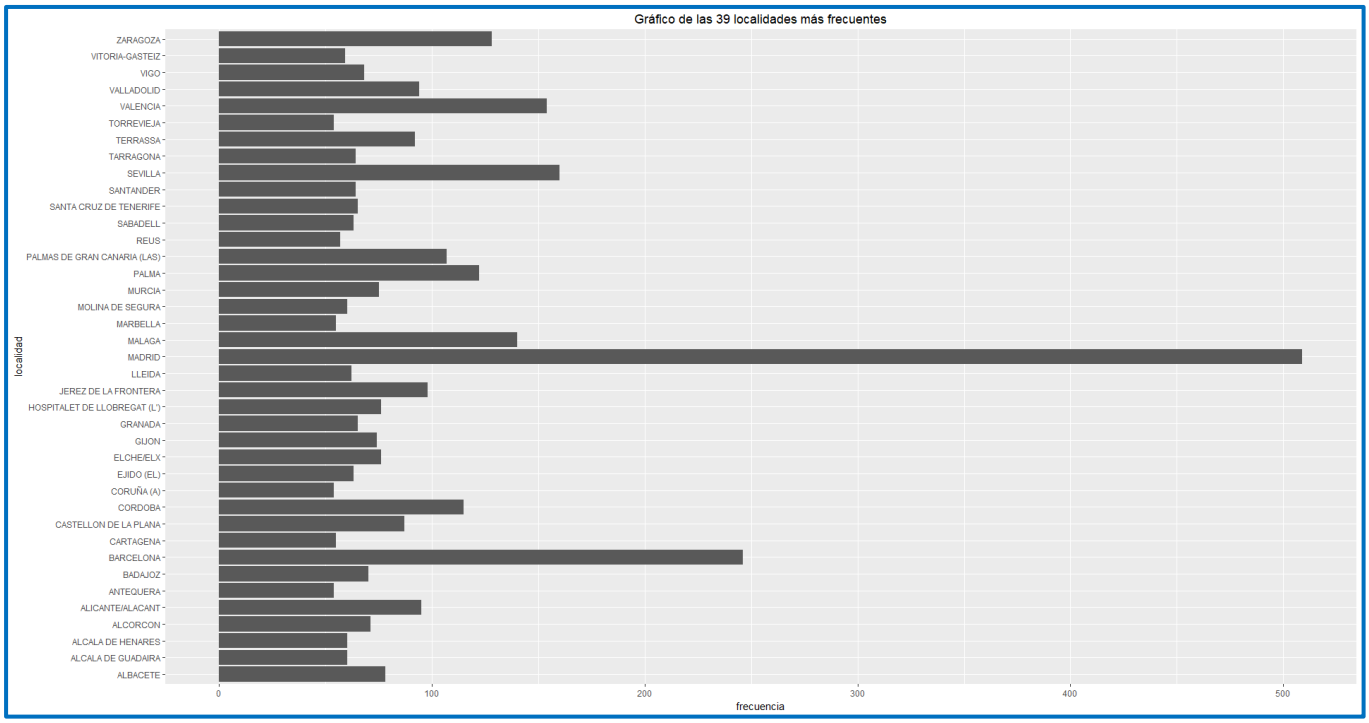


Figura 9: Gráfico de las 39 localidades más frecuentes

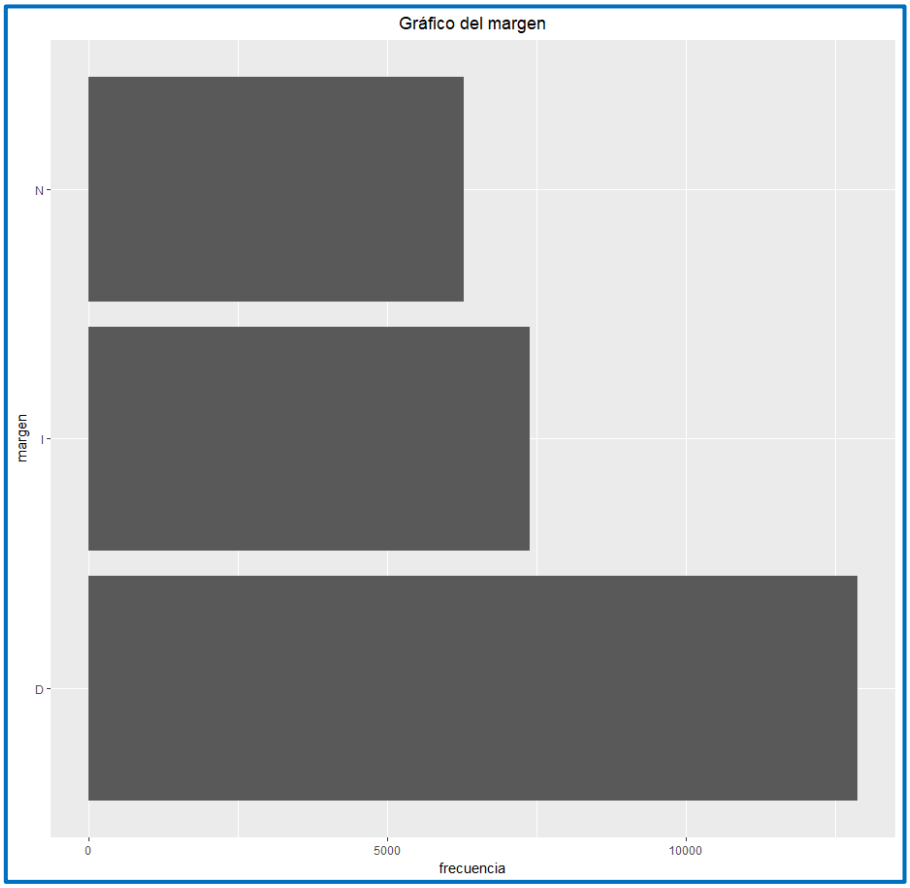


Figura 10: Gráfico del margen

```
> MunicipioMasFrecuente = head(sort(table(gasolineras$municipio), decreasing=T), 47)
> MunicipioMasFrecuente

                                Madrid                                 Murcia                              Barcelona
                                   509                                    266                                    246
                              Valencia                              Cartagena                                Sevilla
                                   170                                    163                                    162
                              Zaragoza                         Palma de Mallorca                                Málaga
                                   160                                    152                                    147
                               Córdoba                 Palmas de Gran Canaria (Las)                  Alicante/Alacant
                                   136                                    119                                    111
                 Jerez de la Frontera                              Elche/Elx                             Valladolid
                                   109                                    107                                     96
                              Terrassa                                Gijón                            Ejido (El)
                                    92                                     89                                     88
Castellón de la Plana/Castelló de la Plana                             Badajoz                              Albacete
                                    87                                     83                                     82
                       Vitoria-Gasteiz                 Hospitalet de Llobregat (L')                              Vigo
                                    77                                     76                                     74
                                 Lorca                                Alcorcón                             Tarragona
                                    72                                     71                                     70
                                Lleida                              Marbella                 Santa Cruz de Tenerife
                                    68                                     68                                     68
                               Almería                           Coruña (A)                               Granada
                                    66                                     65                                     65
                              Orihuela                             Santander                              Sabadell
                                    65                                     64                                     63
                  Alcalá de Guadaíra                        Alcalá de Henares                                  Lugo
                                    62                                     60                                     60
                       Molina de Segura                                Telde                              Estepona
                                    60                                     59                                     58
                             Antequera                                 Reus                        Dos Hermanas
                                    57                                     57                                     56
               San Cristóbal de La Laguna                      Sagunto/Sagunt
                                    56                                     55
```

Figura 11: Municipios más frecuentes

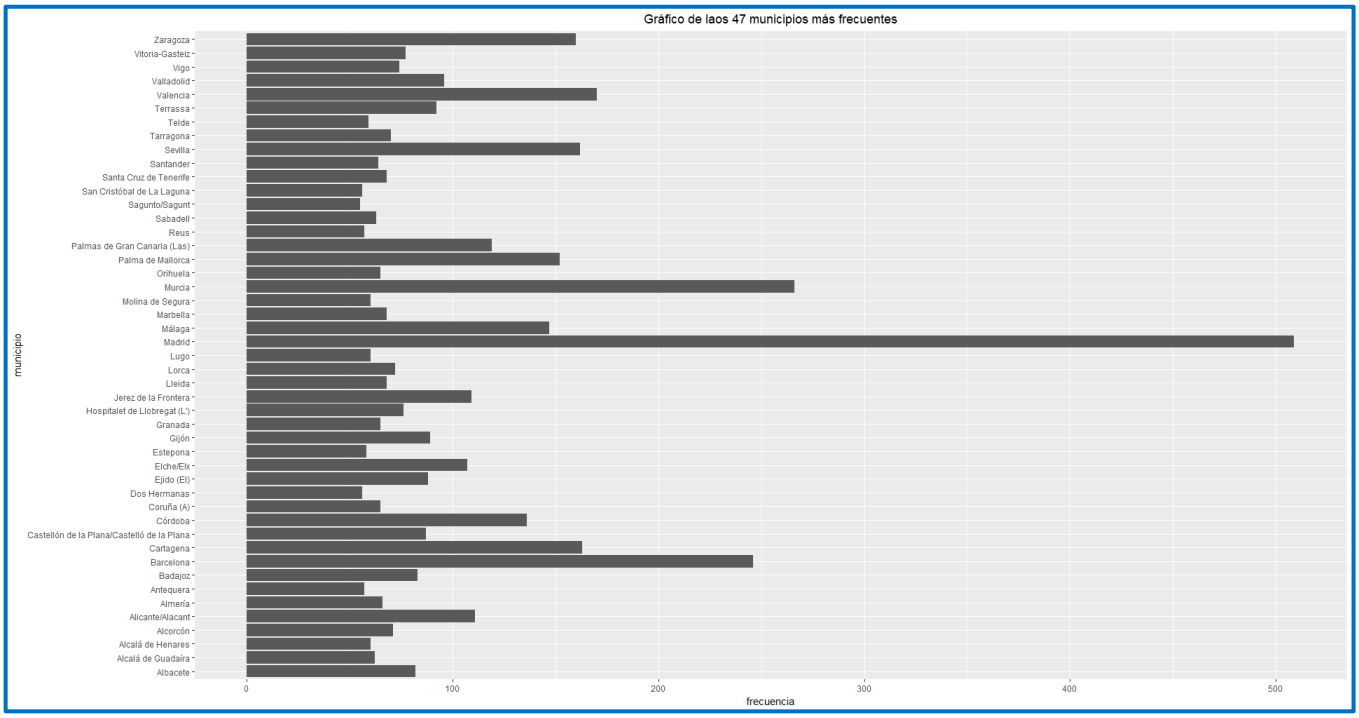


Figura 12: Gráfico de los 47 municipios más frecuentes

```
> RotuloMasFrecuente = head(sort(table(gasolineras$rotulo), decreasing=T), 20)
> RotuloMasFrecuente

   REPSOL     CEPSA      GALP     SHELL        BP  PETRONOR CAMPSA CARREFOUR      AVIA BALLENOIL    MEROIL     SARAS    EROSKI      AGLA   BONAREA
     7982      3830      1430       938       589       587     527       360       343       172       153       126       117       102       102
 VALCARCE  IBERDOEX PETROPRIX ESCLATOIL  PETROCAT
       89        83        76        74        73
```

Figura 13: Nombres más frecuentes

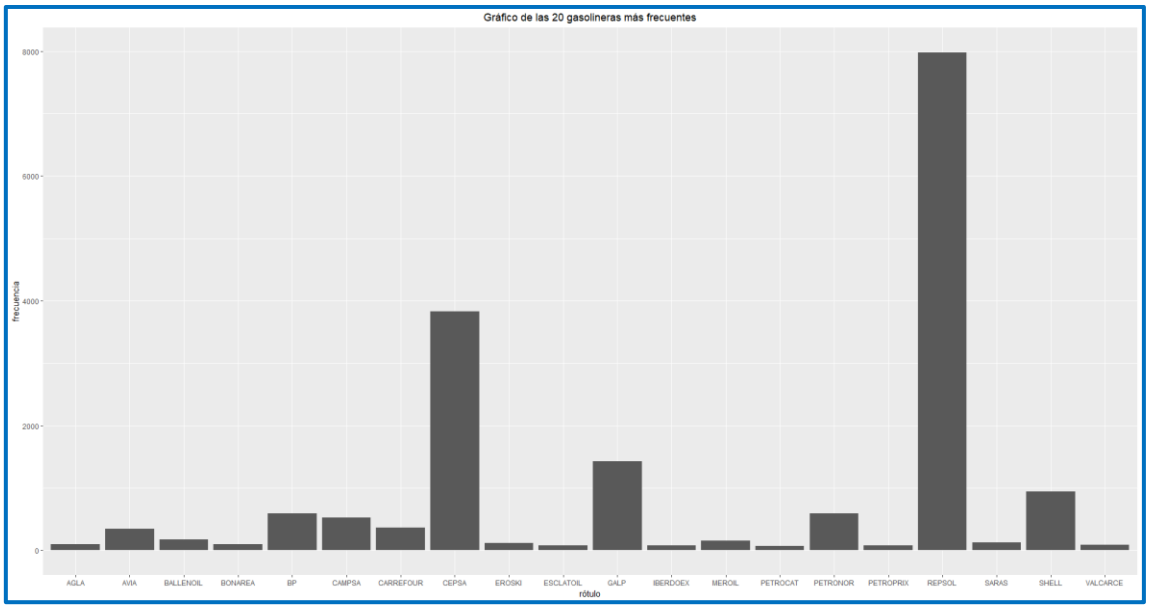


Figura 14: Gráfico de los 20 nombres de las gasolineras más frecuentes

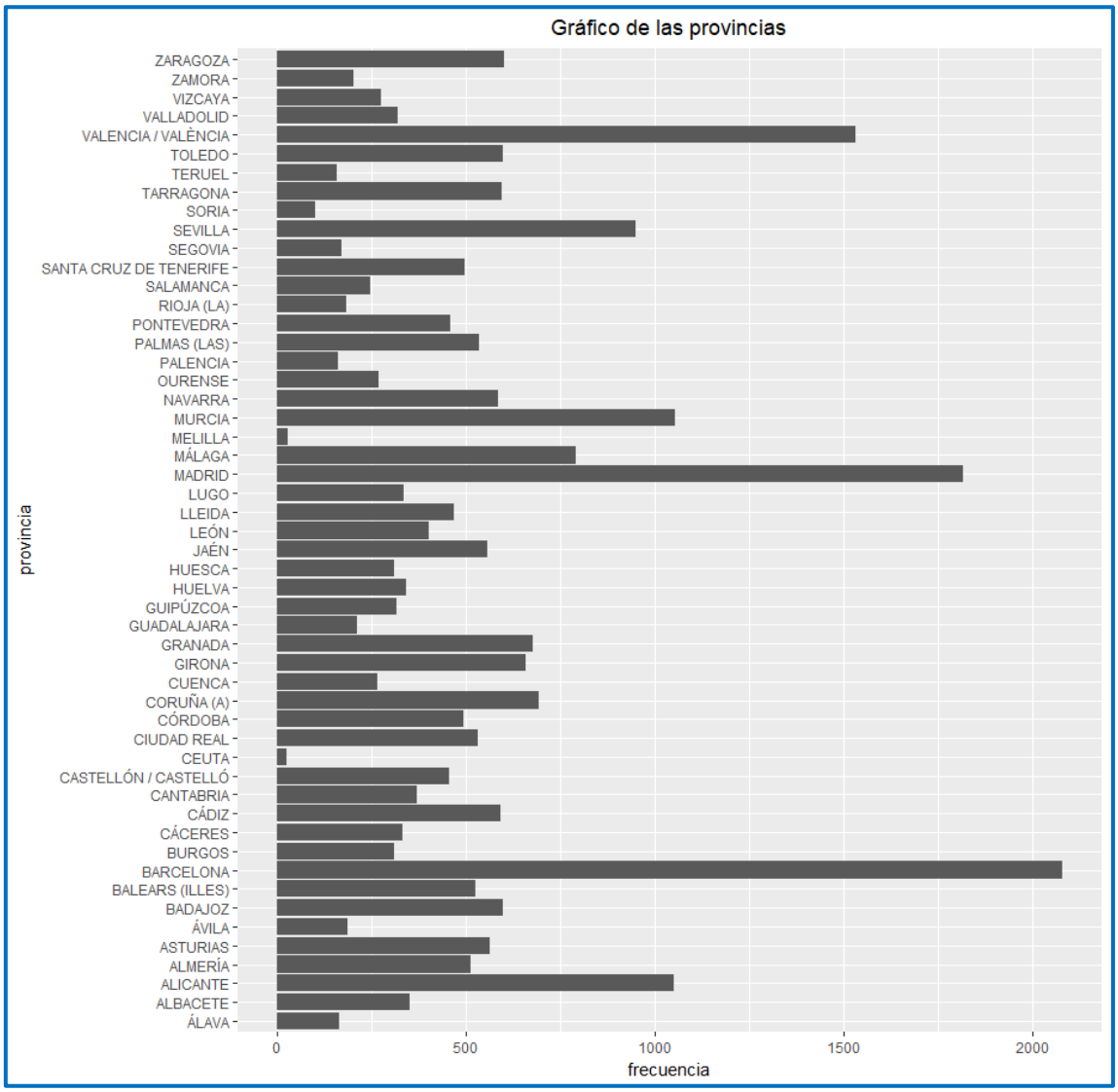


Figura 15: Gráfico de las provincias

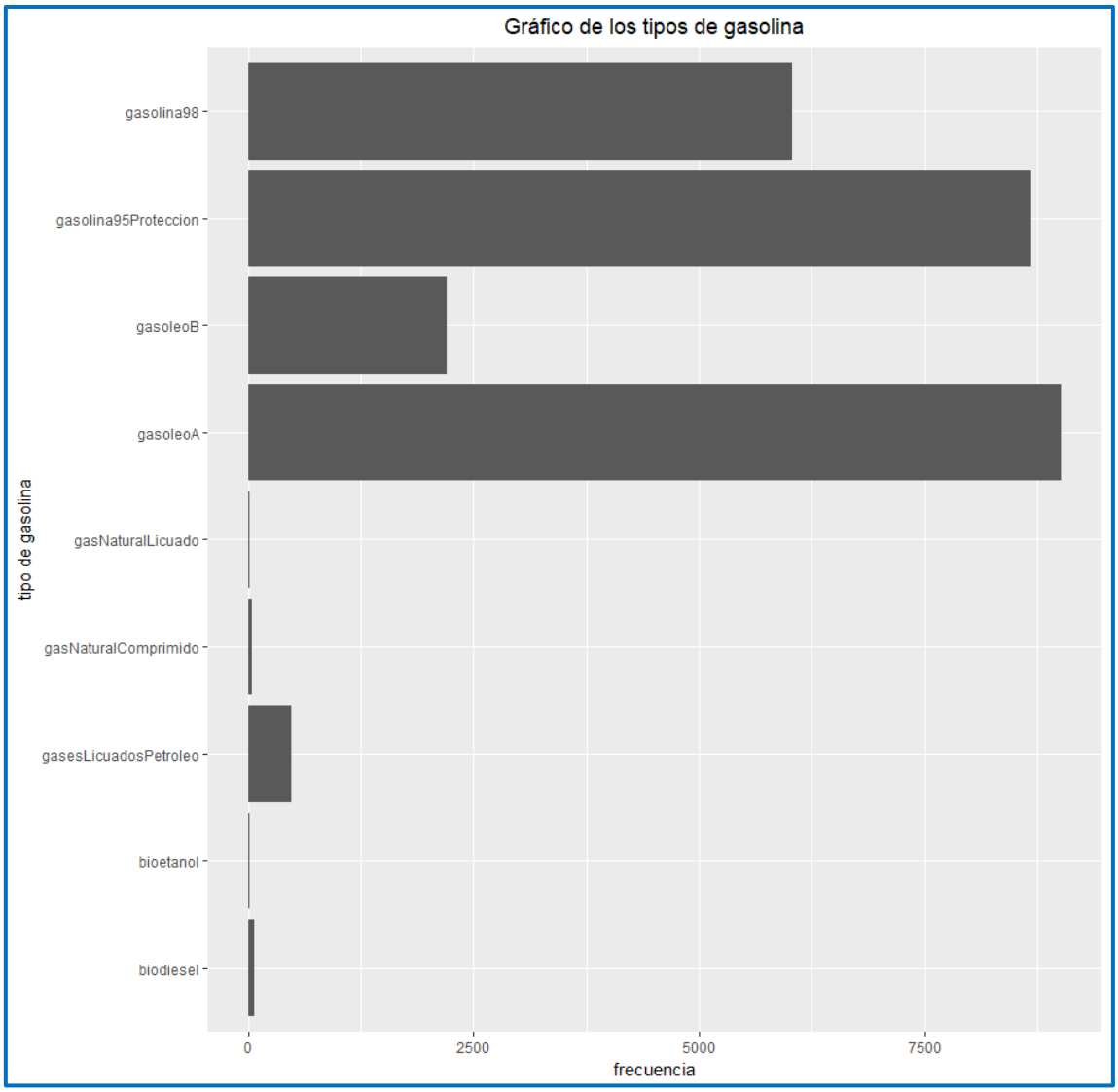


Figura 16: Gráfico del tipo de gasolina

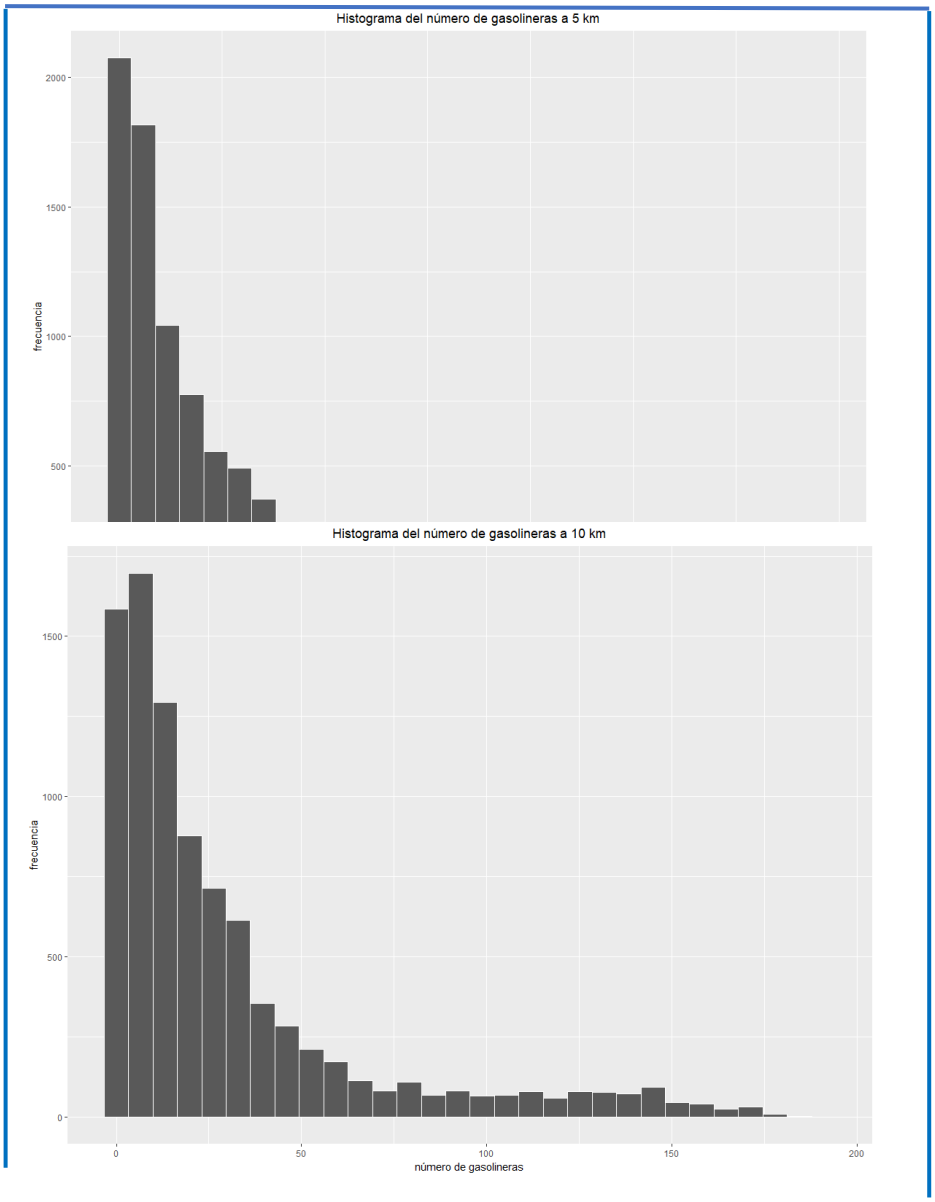
B Anexo descriptivo de las variables de la competencia

- ***Para el precio medio***,

Gasóleo A

Precio por radio	Media del precio medio
Precio 5 km	1.131645
Precio 10 km	1.131787
Precio 20 km	1.132142
Precio 50 km	1.131882

Tabla 1: Media del precio medio por radio de gA



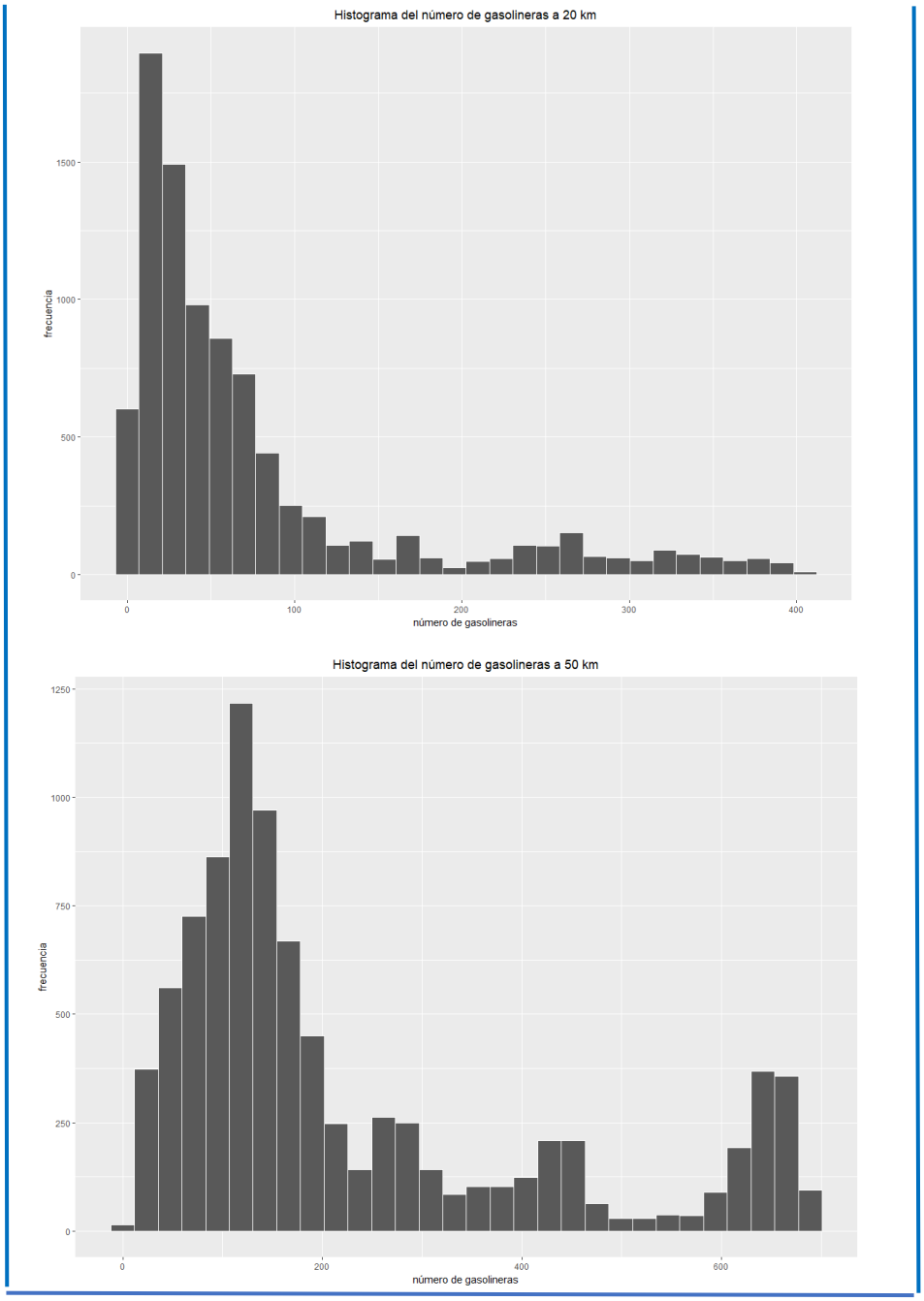


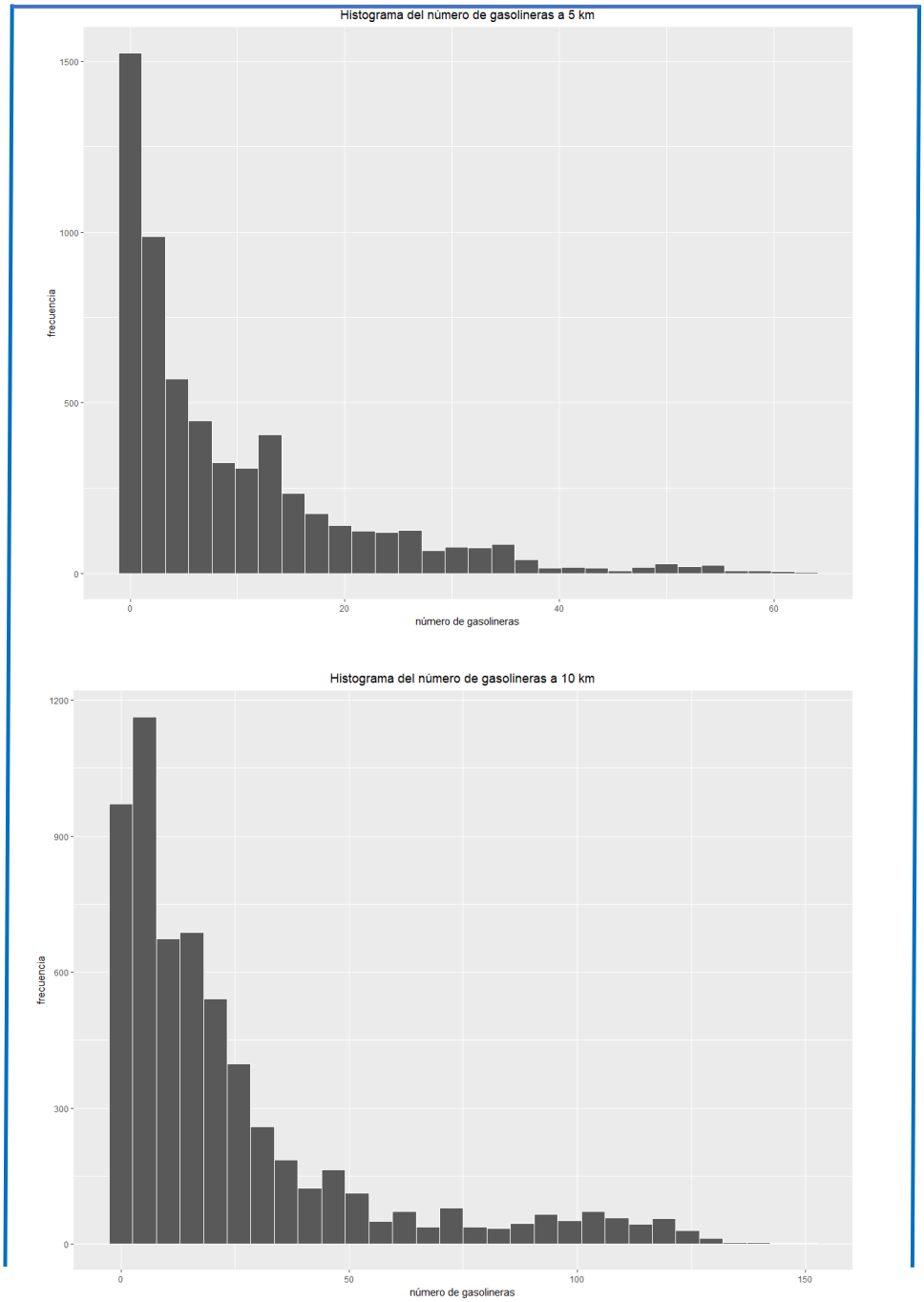
Figura 33: Histogramas del número de gasolineras del precio medio por radio de gA

Gasolina 98

Precio por radio	Media de la media del precio
Precio 5 km	1.146103
Precio 10 km	1.146226
Precio 20 km	1.146358
Precio 50 km	1.146281

Tabla 2: Media del precio medio por radio de g98

:



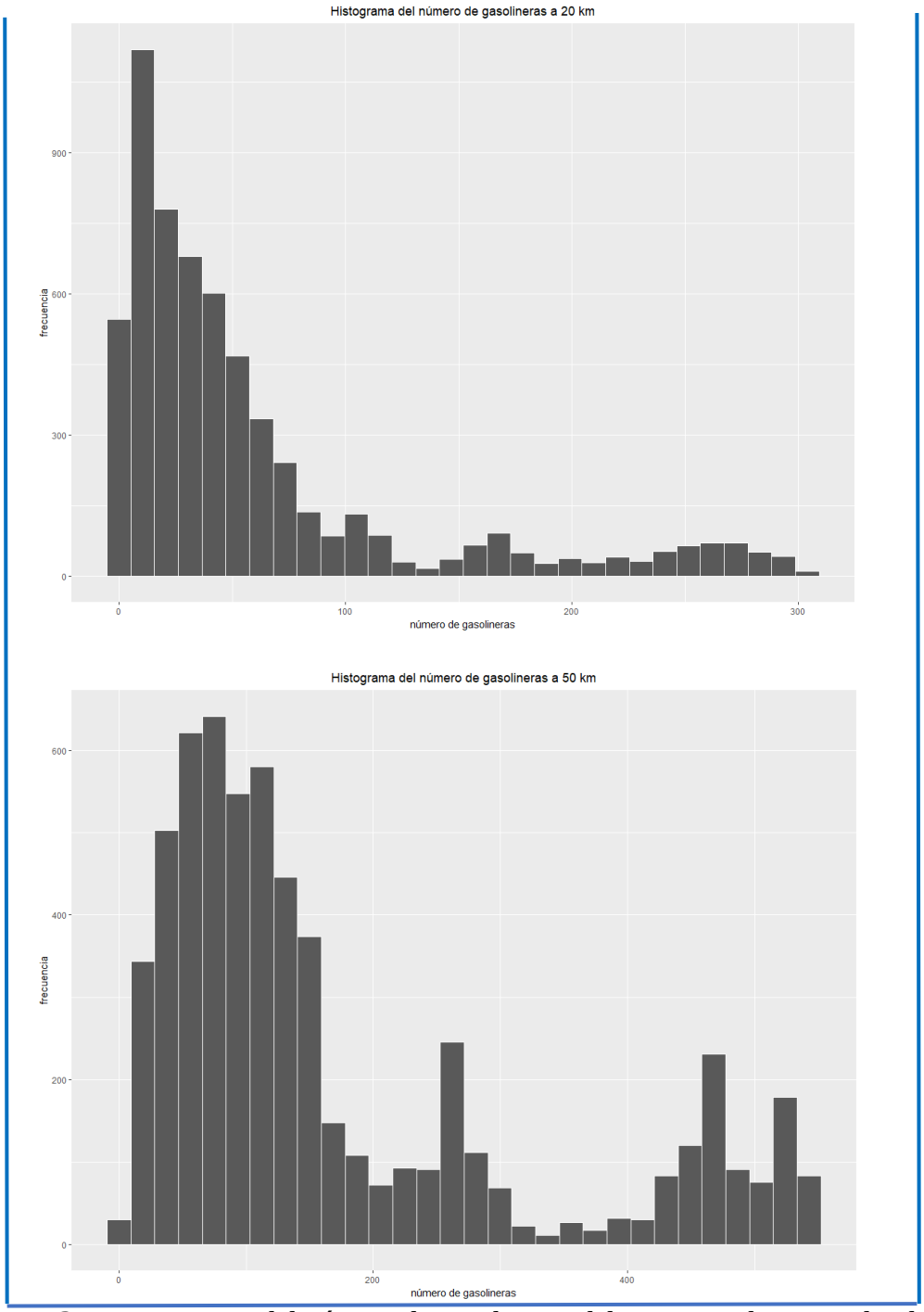


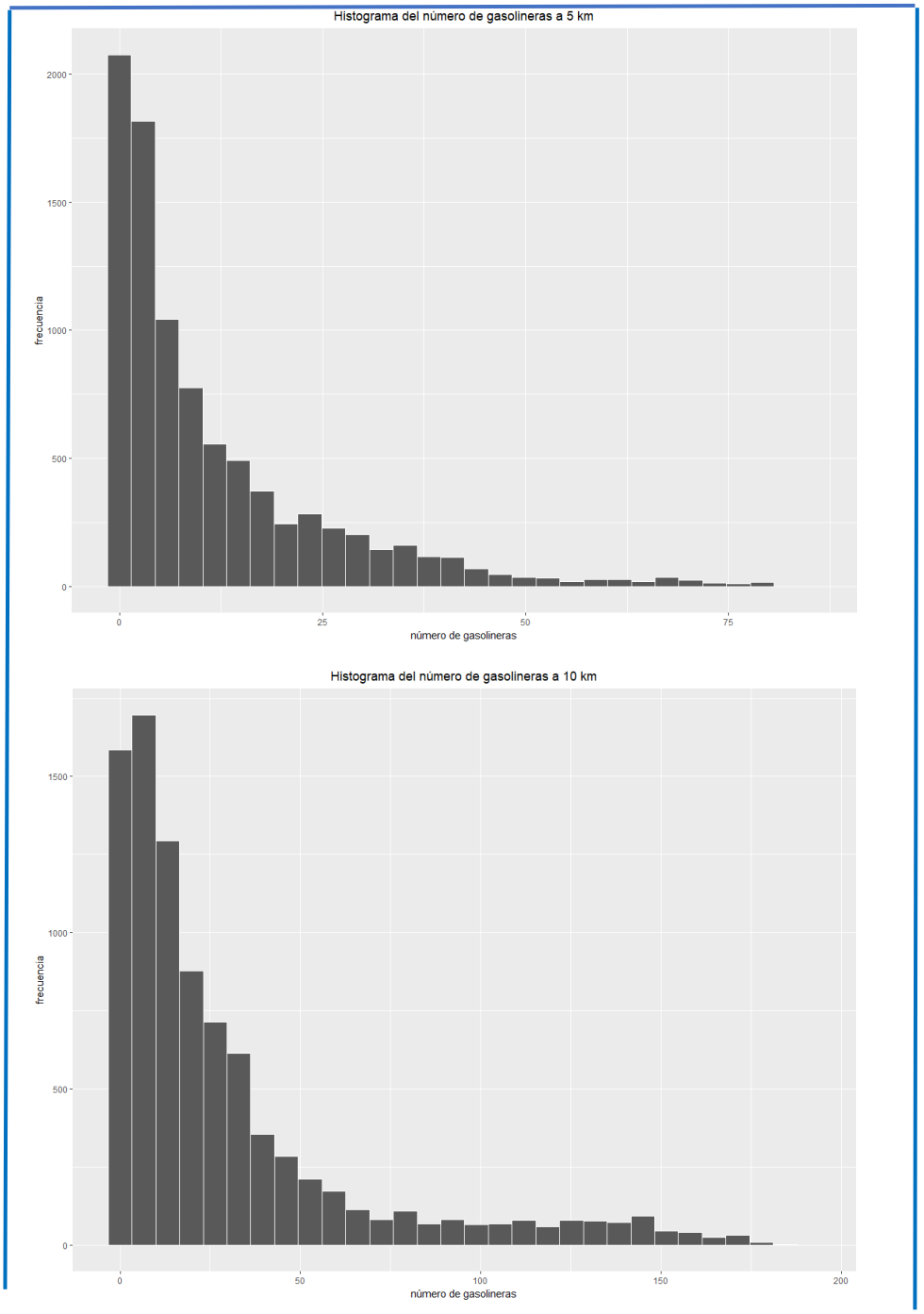
Figura 34: Histogramas del número de gasolineras del precio medio por radio de g98

- ***<u>Para el precio mínimo</u>,***

Gasóleo A

Precio por radio	Media del precio mínimo
Precio 5 km	1.049887
Precio 10 km	1.030727
Precio 20 km	1.008861
Precio 50 km	0.984272

Tabla 3: Media del precio mínimo por radio de gA



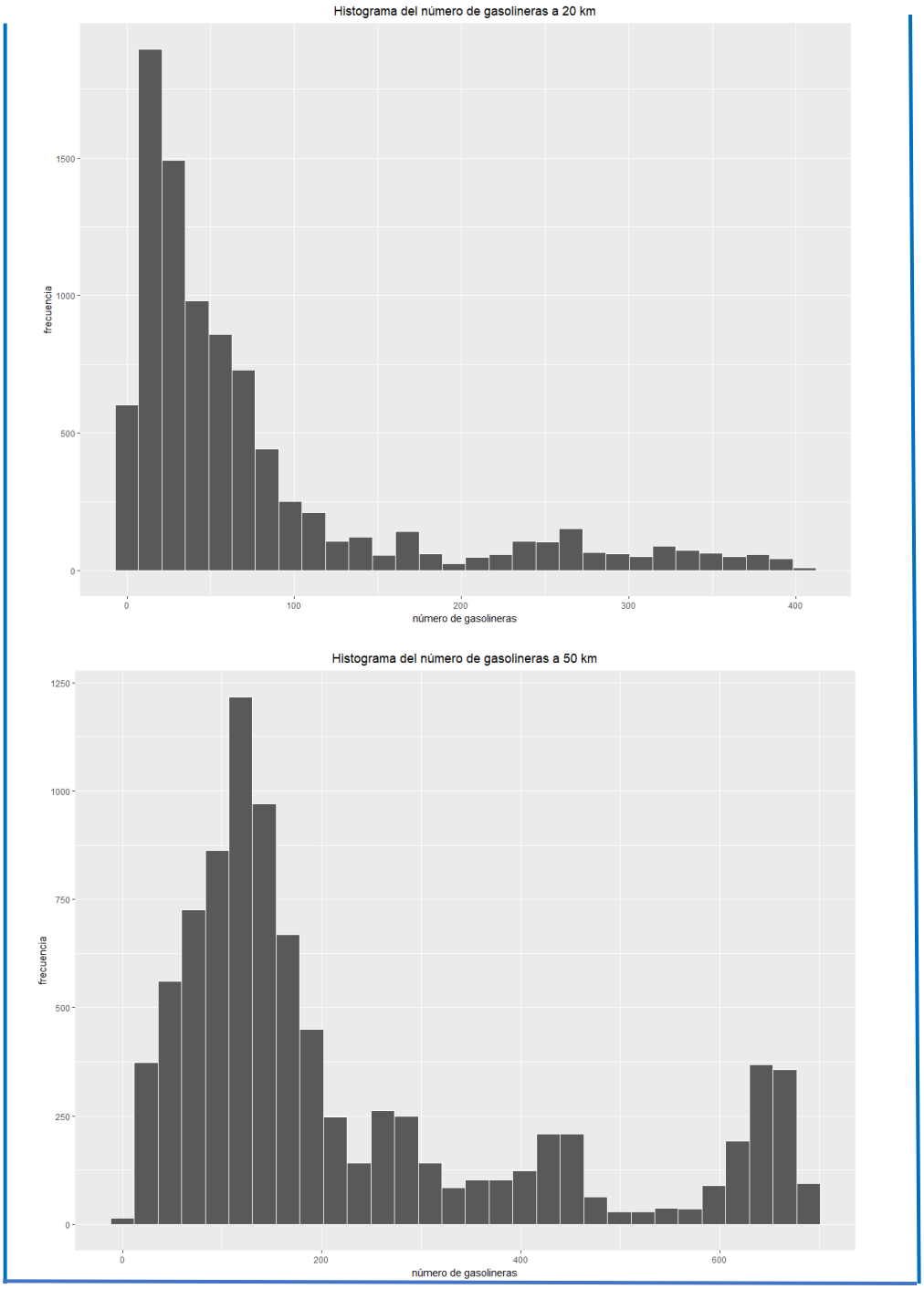
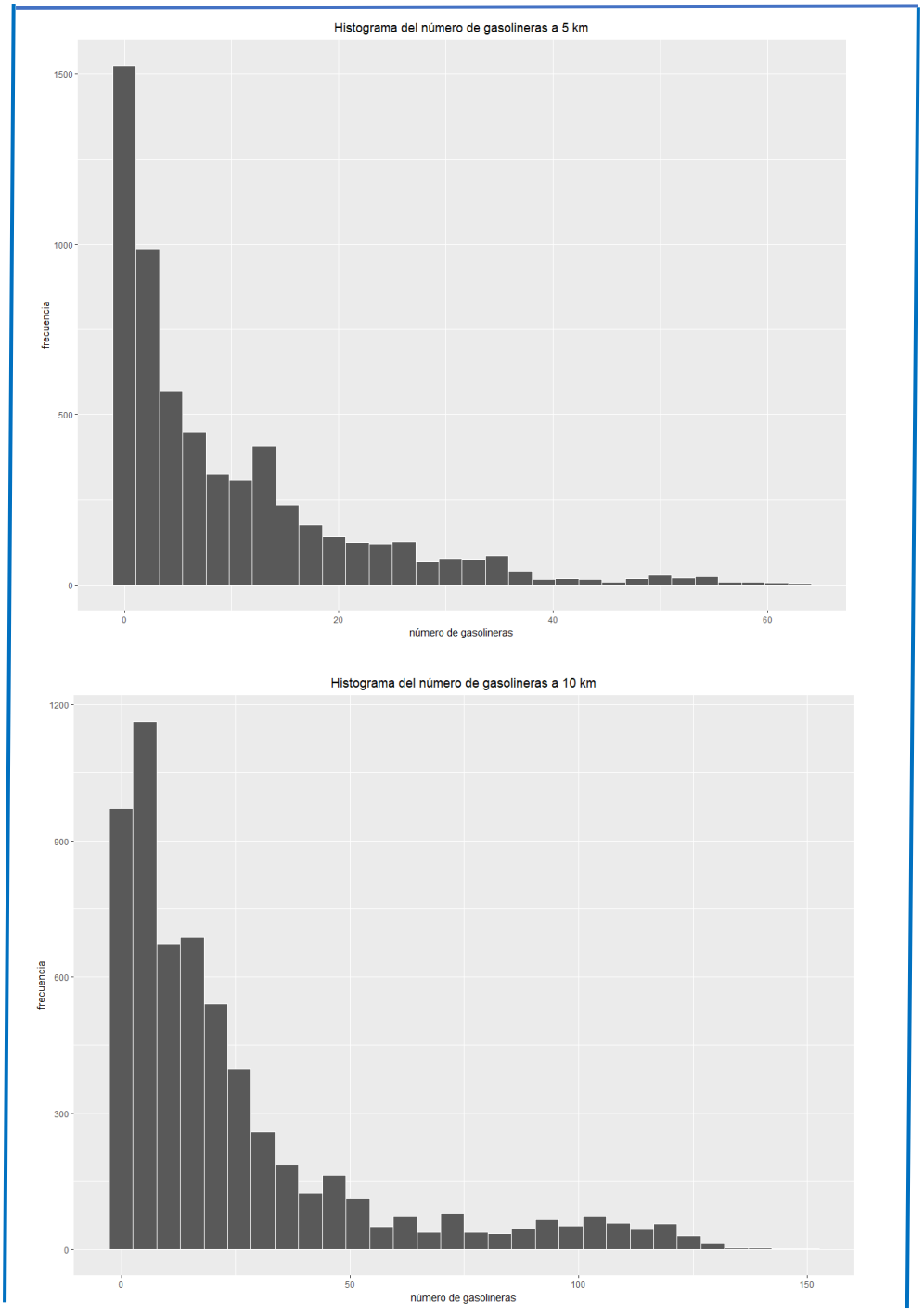


Figura 35: Histogramas del número de gasolineras del precio mínimo por radio de gA

Gasolina 98

Precio por radio	Media del precio mínimo
Precio 5 km	1.076075
Precio 10 km	1.057459
Precio 20 km	1.035816
Precio 50 km	1.006051

Tabla 4: Media del precio mínimo por radio de g98



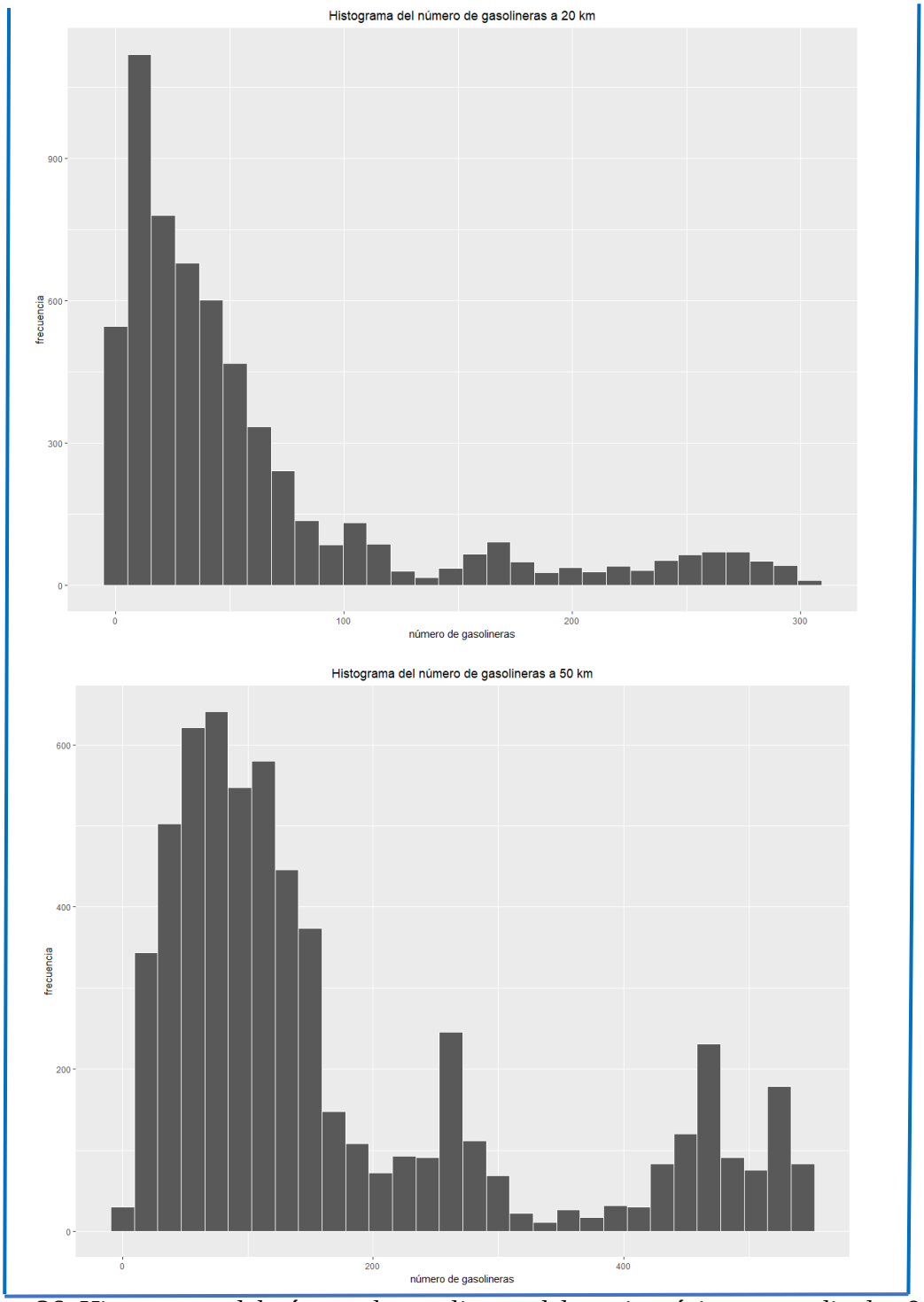


Figura 36: Histogramas del número de gasolineras del precio mínimo por radio de g98

C Anexo descriptivo de los valores atípicos de precio en función del tipo de gasolina según el Test de Grubbs

Gasóleo A

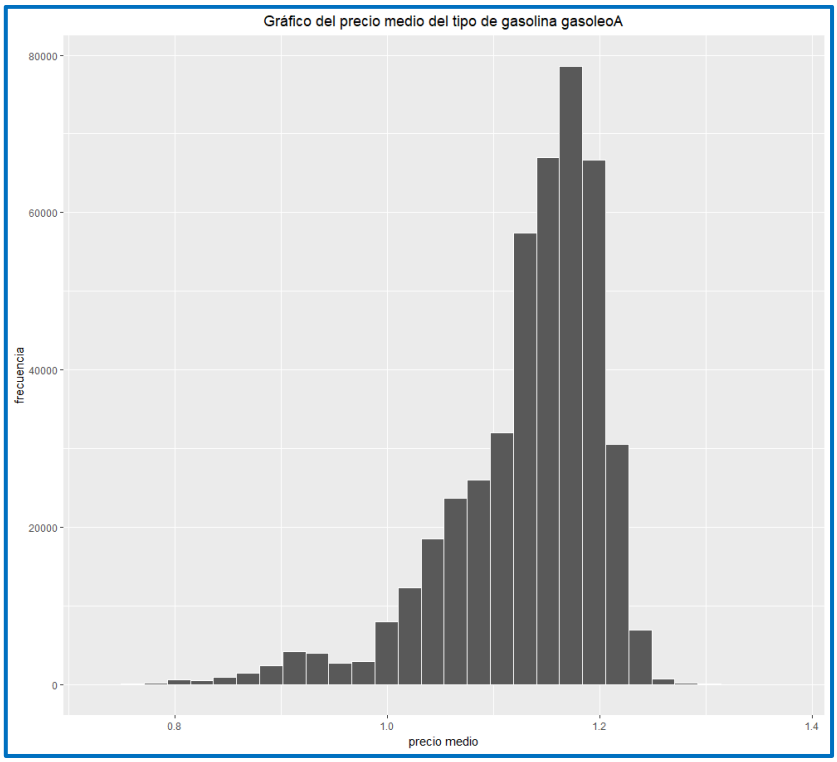


Figura 49: Histograma del precio medio de Gasóleo A

Gas Natural Comprimido

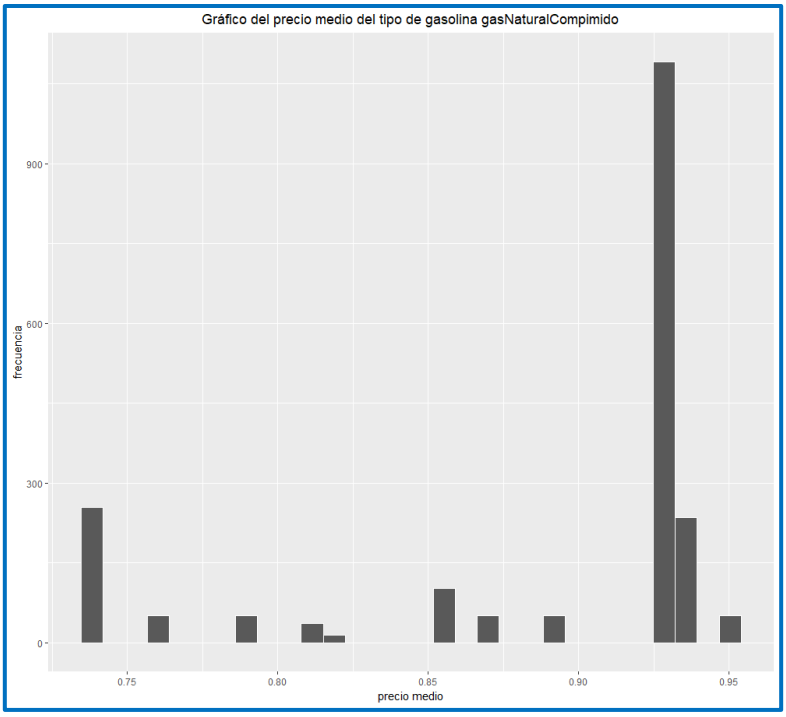


Figura 50: Histograma del precio medio de GN

Gases Licuados Petróleo

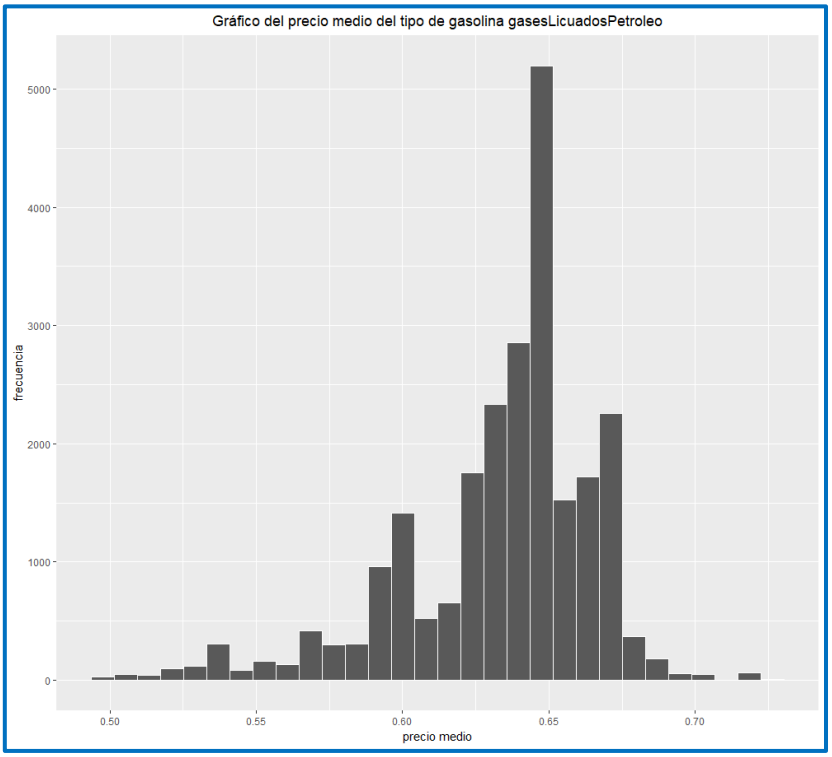


Figura 51: Histograma del precio medio de GLP

Bioetanol

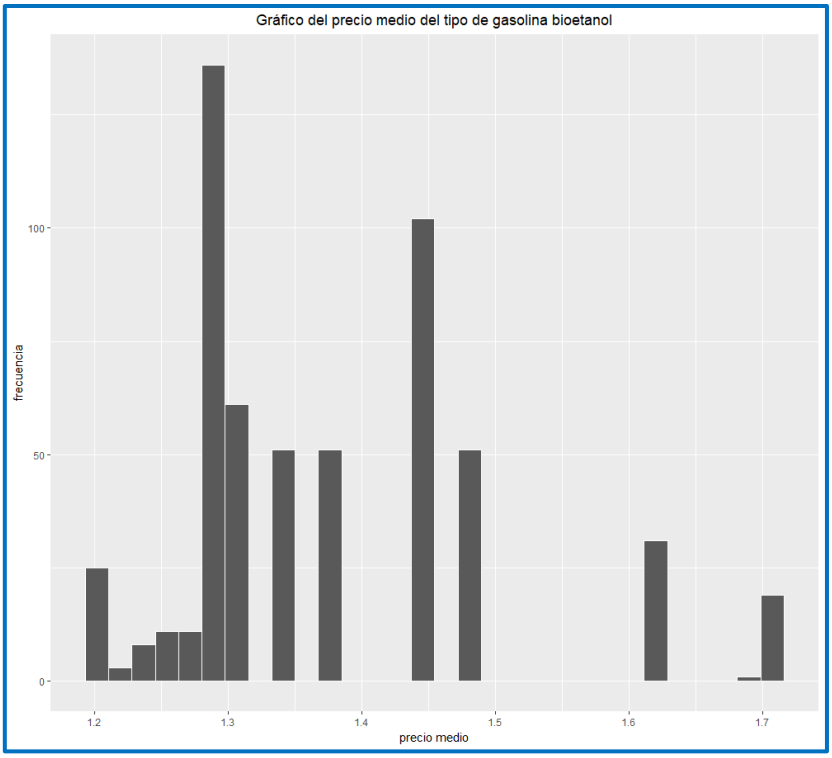


Figura 52: Histograma del precio medio de bioetanol

D Anexo del Análisis Clúster 2

***Clúster 2:** Precio de la gasolina, y precio mínimo de la competencia a 5, 10, 20, y 50 km*

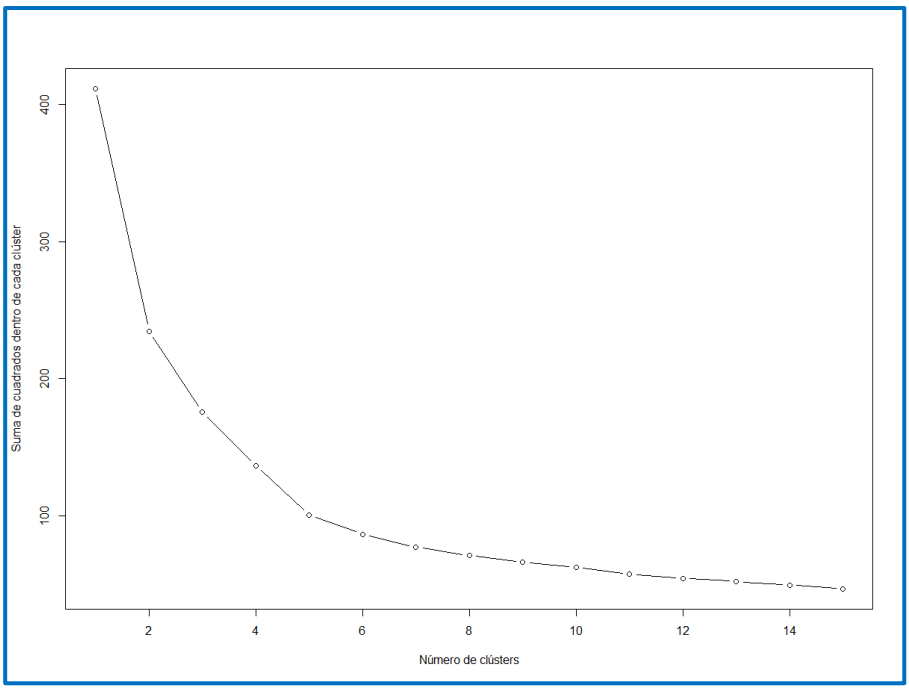


Figura 60: Gráfico del número óptimo de clúster en el Clúster 2

- *La agrupación de las gasolineras de España en 5 clústeres es:*


precio_5km
media_precio
precios_preciosCompetencMin_Cluster$cluster
1
2
3
4
5


precio_10km
media_precio
precios_preciosCompetencMin_Cluster$cluster
1
2
3
4
5

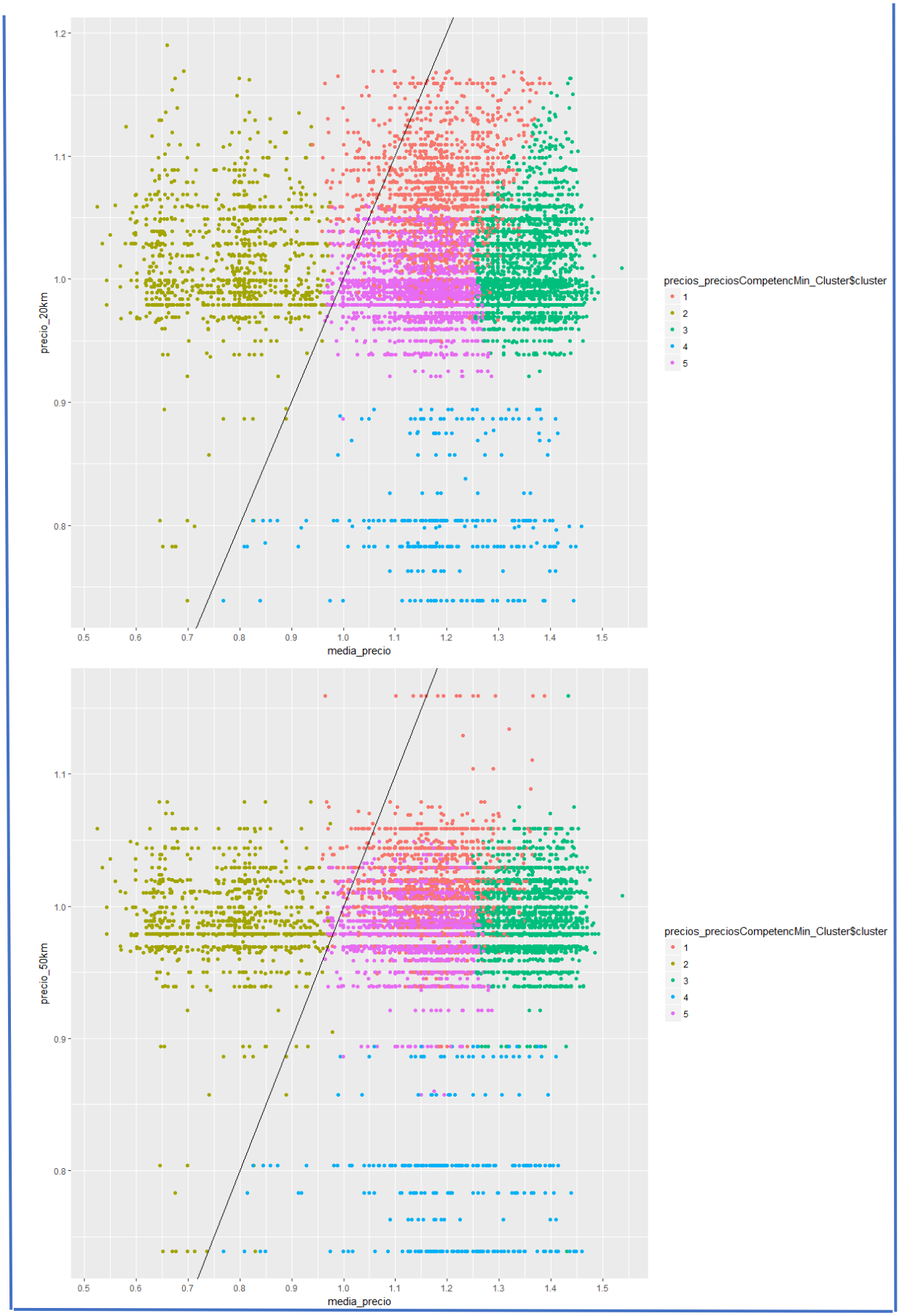


Figura 61: Clúster 2 de los precios de la gasolina y mínimos de la competencia

- *El número de gasolineras y el gráfico de las mismas por provincia y por conglomerado es:*

```
> precios_preciosCompetencMin_Cluster$size
[1]  981  318 1955 3142 2621
```

Figura 62: Número de gasolineras por clúster del Clúster 2

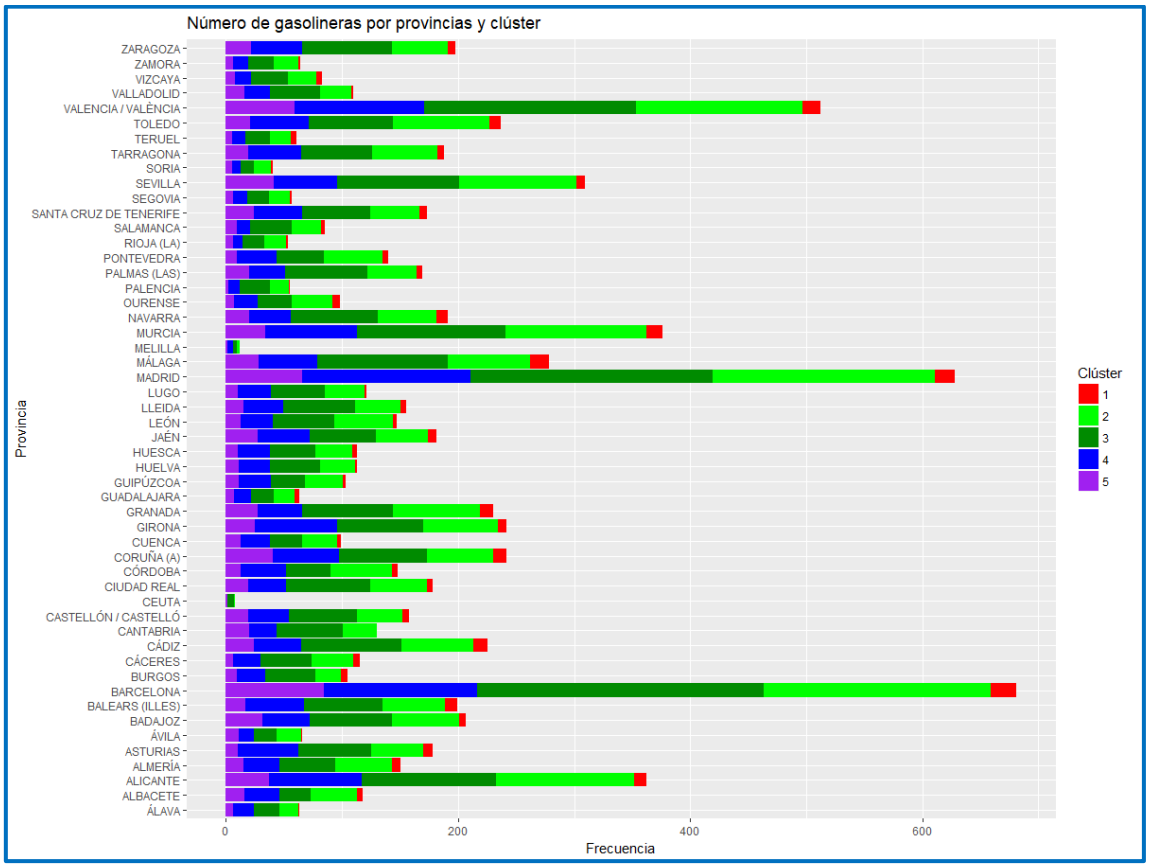


Figura 63: Gráfico del número de gasolineras por provincias y clúster del Clúster 2

E Anexo del modelo predictivo de Regresión Lineal

- *Regresión lineal con todas las variables,*

```
> modelo <- lm(media_precio~.-1, data = precios2)
> modelo

Call:
lm(formula = media_precio ~ . - 1, data = precios2)

Coefficients:
                        latitud                       longitud                      rotuloSHELL
                       0.108932                       0.080597                      -0.364782
     horarioabierto las 24 horas                  rotuloCAMPSA                provinciaBURGOS
                      -0.016533                      -0.212450                      -0.107198
                provinciaMÁLAGA              provinciaZARAGOZA                 provinciaJAÉN
                      -0.576125                      -0.286791                      -0.150876
             provinciaBARCELONA                 provinciaLEÓN                provinciaSEVILLA
                      -0.218812                      -0.208669                      -0.356807
              provinciaGRANADA     provinciaVALENCIA / VALÈNCIA                 provinciaTOLEDO
                      -0.437897                      -0.318142                      -0.810014
               provinciaHUELVA                provinciaBADAJOZ                provinciaPALENCIA
                       0.005266                      -0.191921                       0.025411
               provinciaZAMORA                 provinciaMADRID                  provinciaCUENCA
                      -0.538427                      -0.401557                      -0.084387
            provinciaTARRAGONA                provinciaNAVARRA                   provinciaÁVILA
                      -0.143163                      -0.132610                      -0.105219
          provinciaGUADALAJARA                 provinciaLLEIDA tipo_gasolgasolina95Proteccion
                      -0.252841                      -0.106213                       0.723940
            provinciaALBACETE                   provinciaLUGO tipo_gasolgasesLicuadosPetroleo
                      -0.069286                      -0.483436                       0.286198
       tipo_gasolgasolina98                tipo_gasolgasoleoB
                      -2.364867                       0.827688
```

Figura 66: Variables seleccionadas en RLineal

- *Las variables seleccionadas en cada método de selección de variables son,*

STEPWISE,

```
> variables.seleccion = variables.StepWise
> variables.seleccion
[1] "latitud"    "longitud"   "horario"    "rotulo"     "provincia"  "tipo_gasol"
```

Figura 70: Variables seleccionadas con Stepwise

FORWARD,

```
> variables.seleccion = variables.Forward
> variables.seleccion
[1] "latitud"    "longitud"   "horario"    "rotulo"     "provincia"  "tipo_gasol"
```

Figura 71: Variables seleccionadas con Forward

BACKWARD

```
> variables.seleccion = variables.Backward
> variables.seleccion
[1] "latitud"    "longitud"   "horario"    "rotulo"     "provincia"  "tipo_gasol"
```

Figura 72: Variables seleccionadas con Backward

F Anexo del modelo predictivo de Redes Neuronales

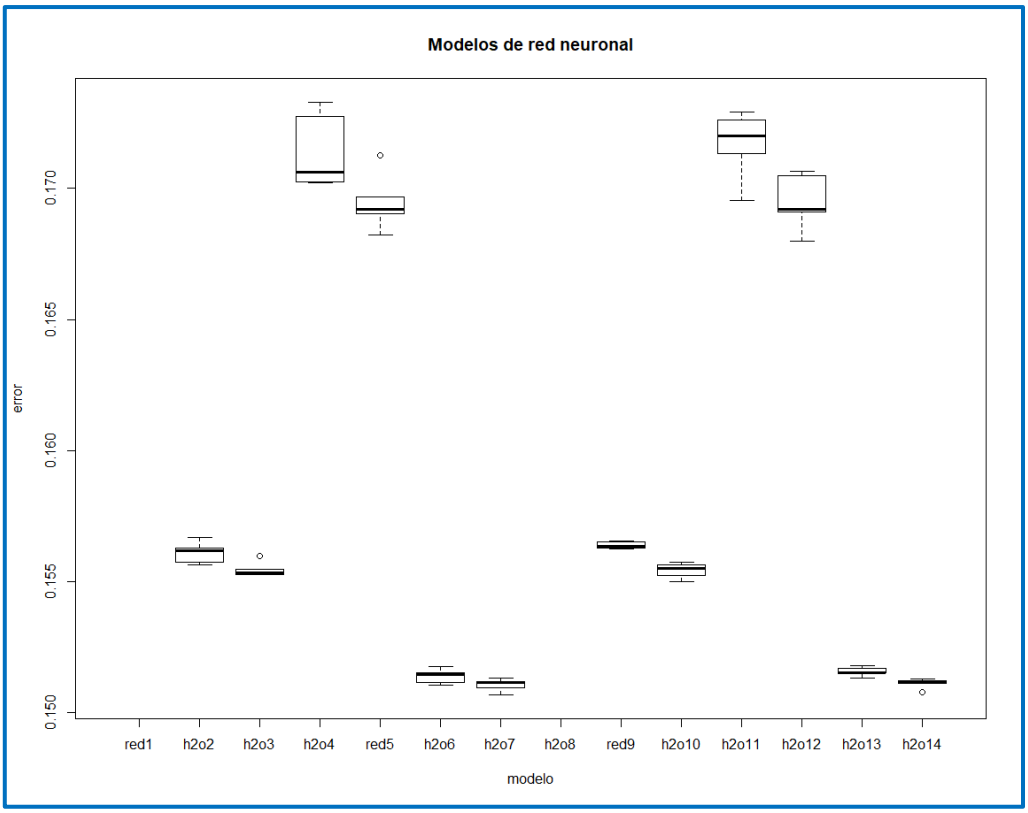


Figura 74: Comparación de modelos de RN

A continuación, separamos por valores similares de error, los modelos de la Figura 74 ya que no se aprecian correctamente. Es decir, los 4 modelos que tienen un error de predicción de 0.17, aproximadamente formarán parte de un mismo gráfico (Figura 76), mientras que los demás, en torno a 0.15 y 0.155, corresponderán a otro gráfico (Figura 77).

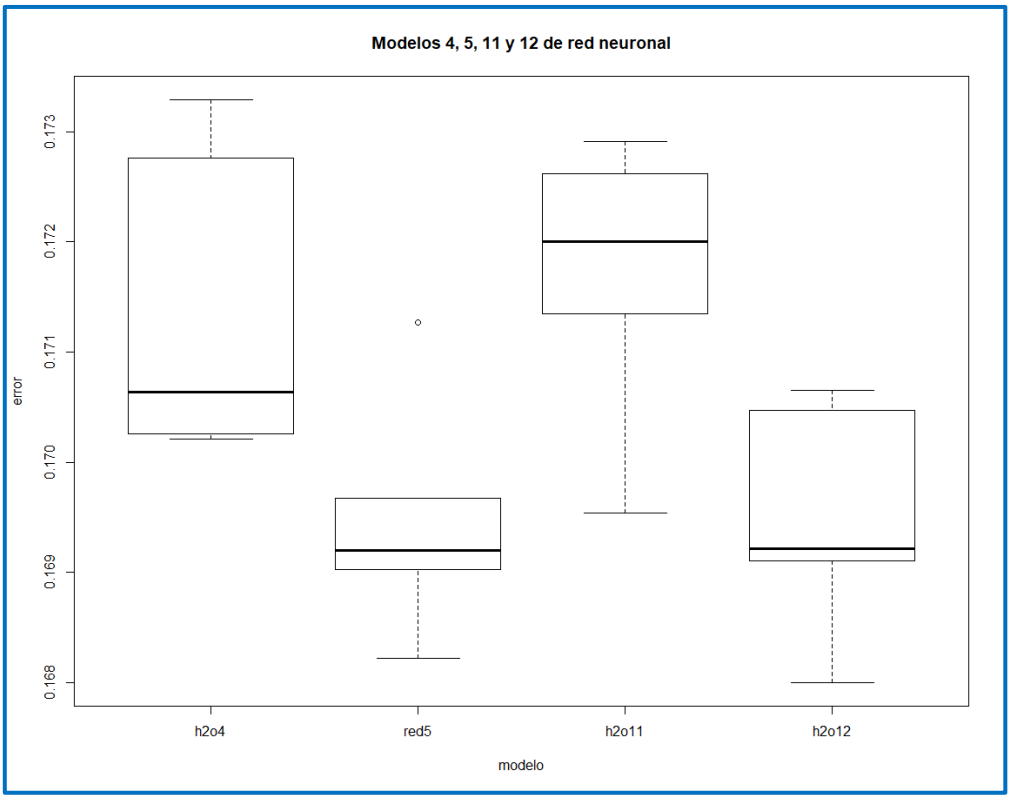


Figura 76: Modelos 4, 5, 11 y 12 de RN

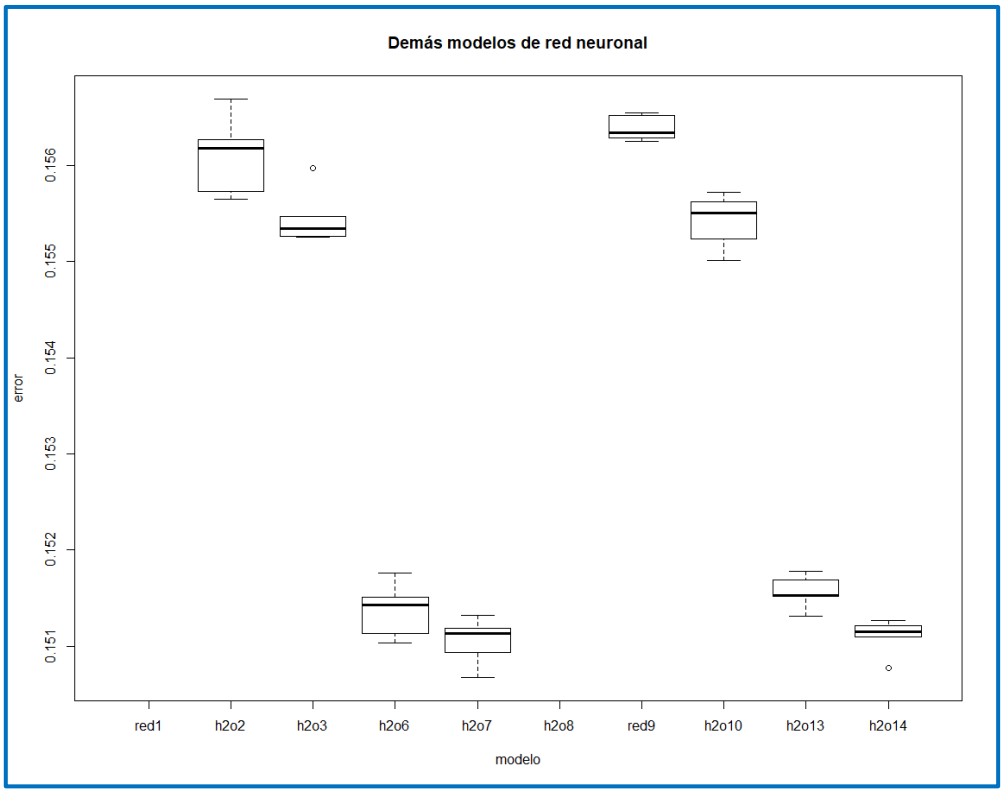


Figura 77: Demás modelos de RN

G Anexo del modelo predictivo de Random Forest

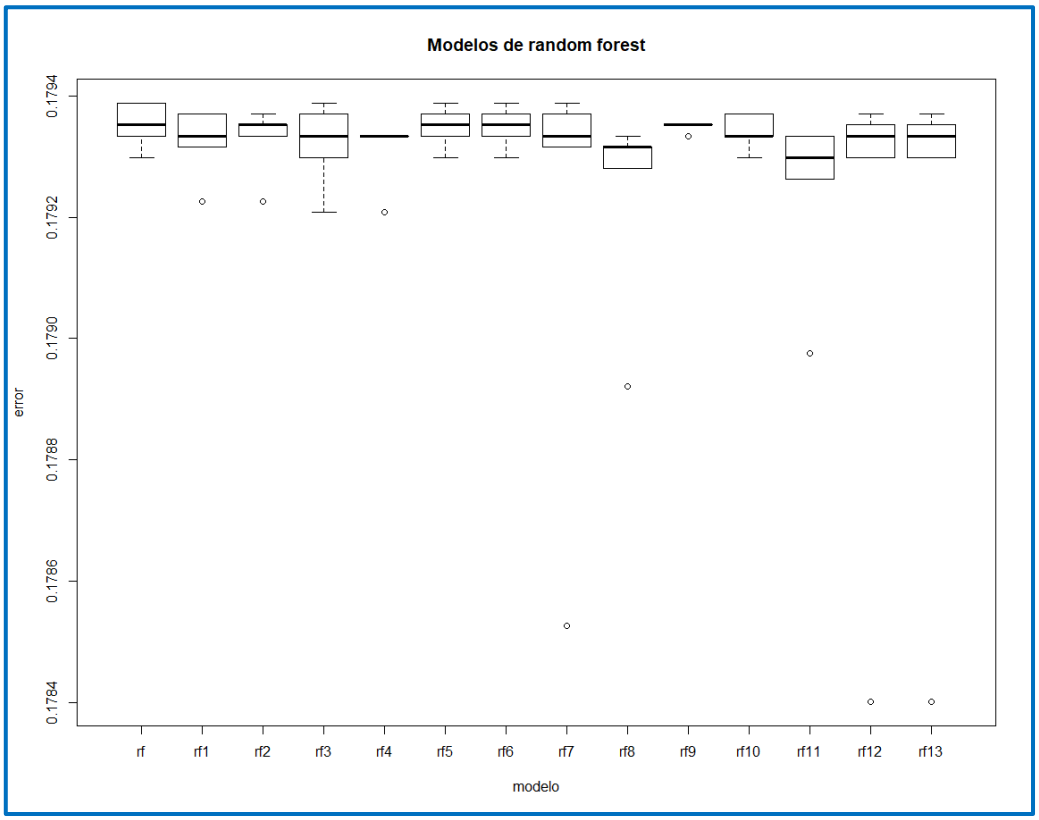


Figura 79: Comparación de modelos de RF

A continuación, separamos por comportamientos similares, los modelos de la Figura 79 ya que no se aprecian correctamente. Es decir, los modelos rf4 y rf9 formarán parte de un mismo gráfico (Figura 81), mientras que los demás, corresponderán a otro gráfico (Figura 82).,

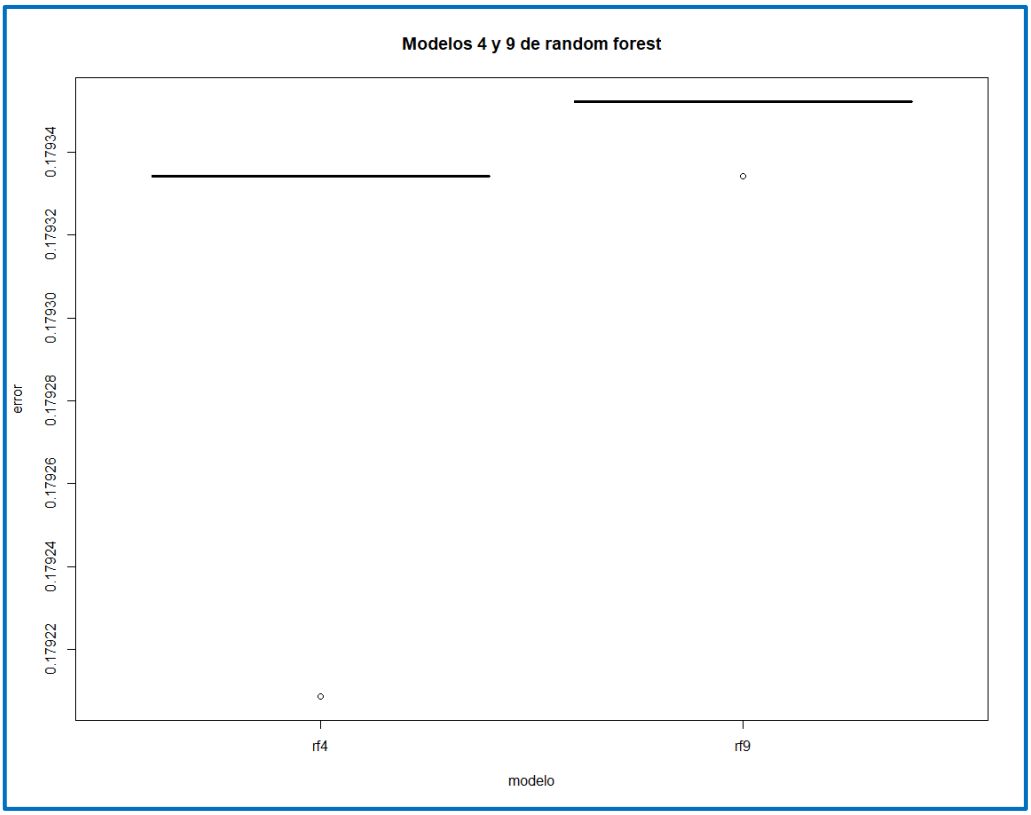


Figura 81: Modelos 1 y 8 de RF

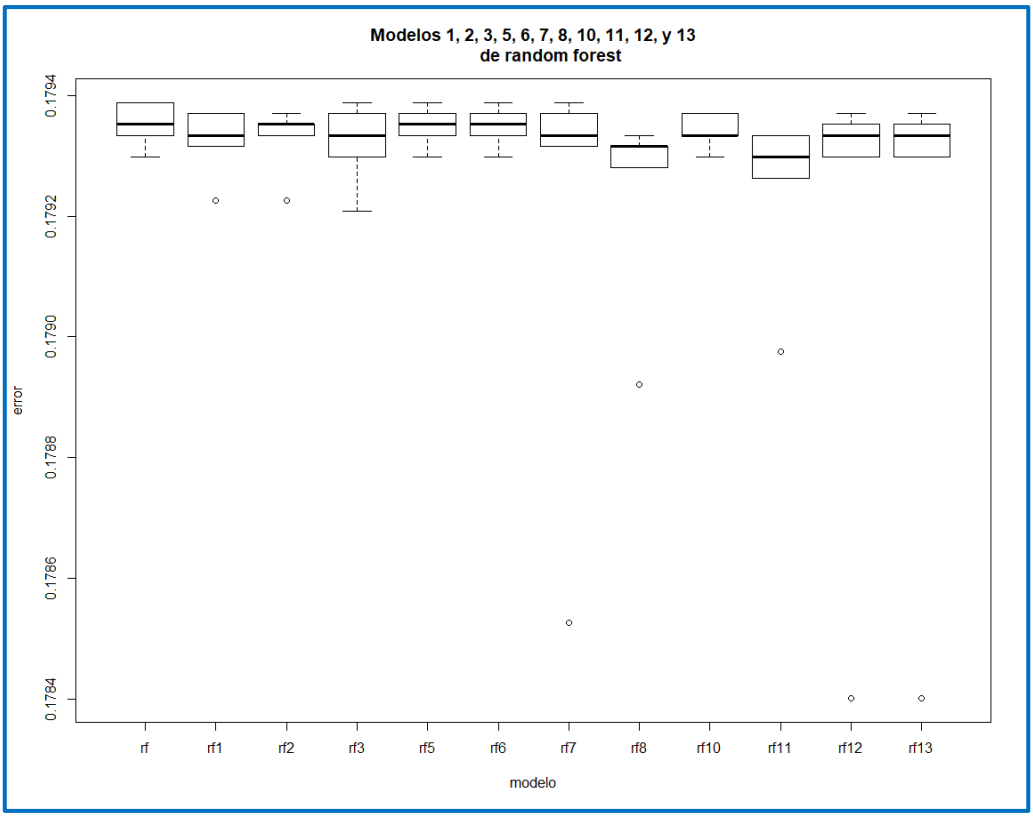


Figura 82: Demás modelos de RF

H Anexo del modelo predictivo de Suport Vector Machine

- *Modelos de SVM con kernel lineal y radial semilla 12346*

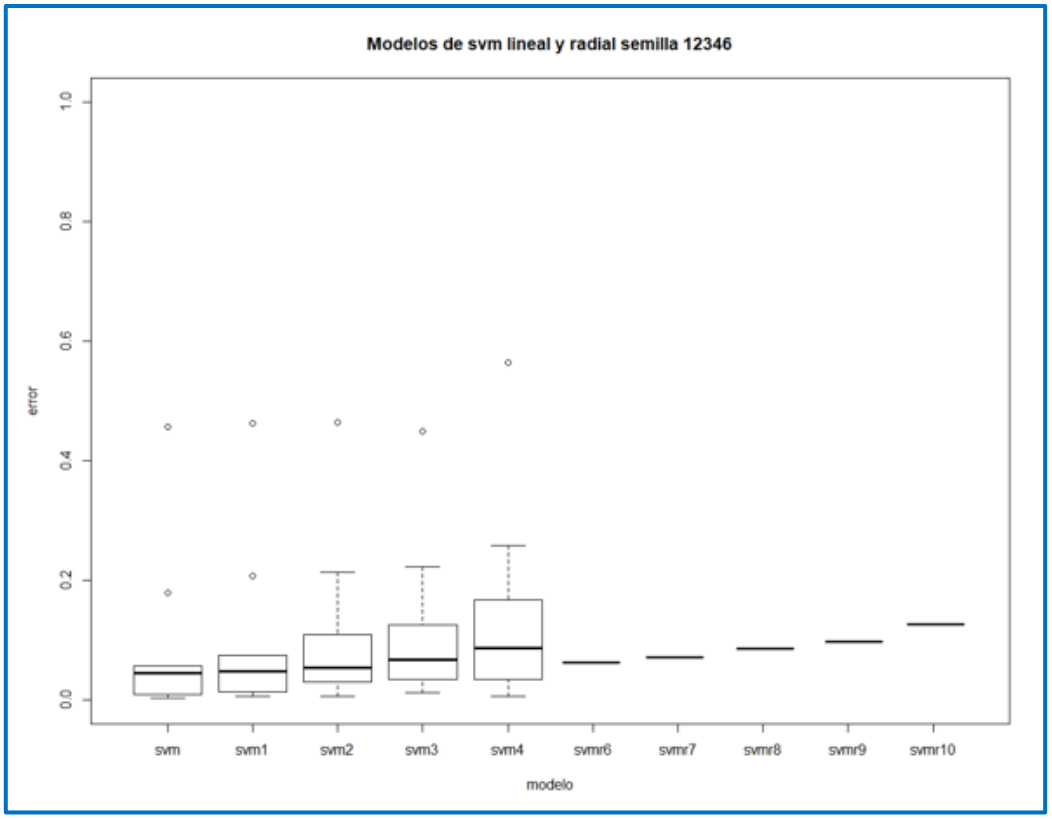


Figura 84: Comparación de modelos de SVM 12346

A continuación, separamos por comportamientos similares, los modelos de la Figura 84 ya que no se aprecian correctamente. Es decir, los modelos $svm - sv4$ formarán parte de un mismo gráfico (Figura 86), mientras que los demás, corresponderán a otro gráfico (Figura 87).,

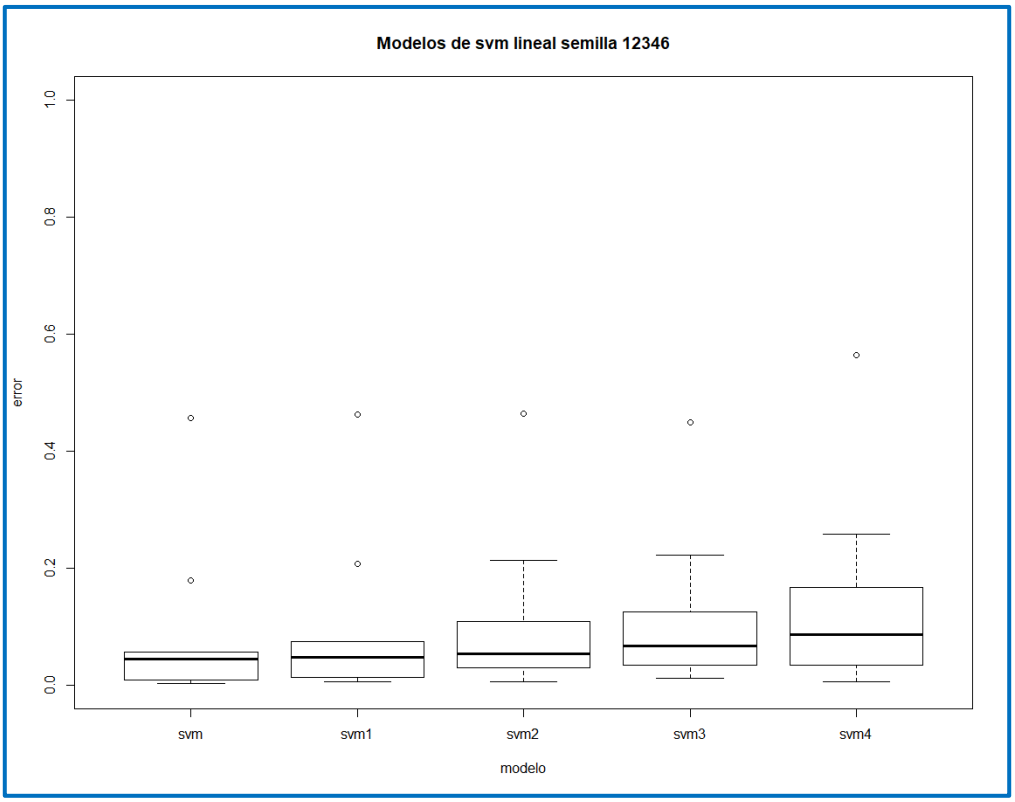


Figura 86: Modelos de SVM con kernel lineal semilla 12346

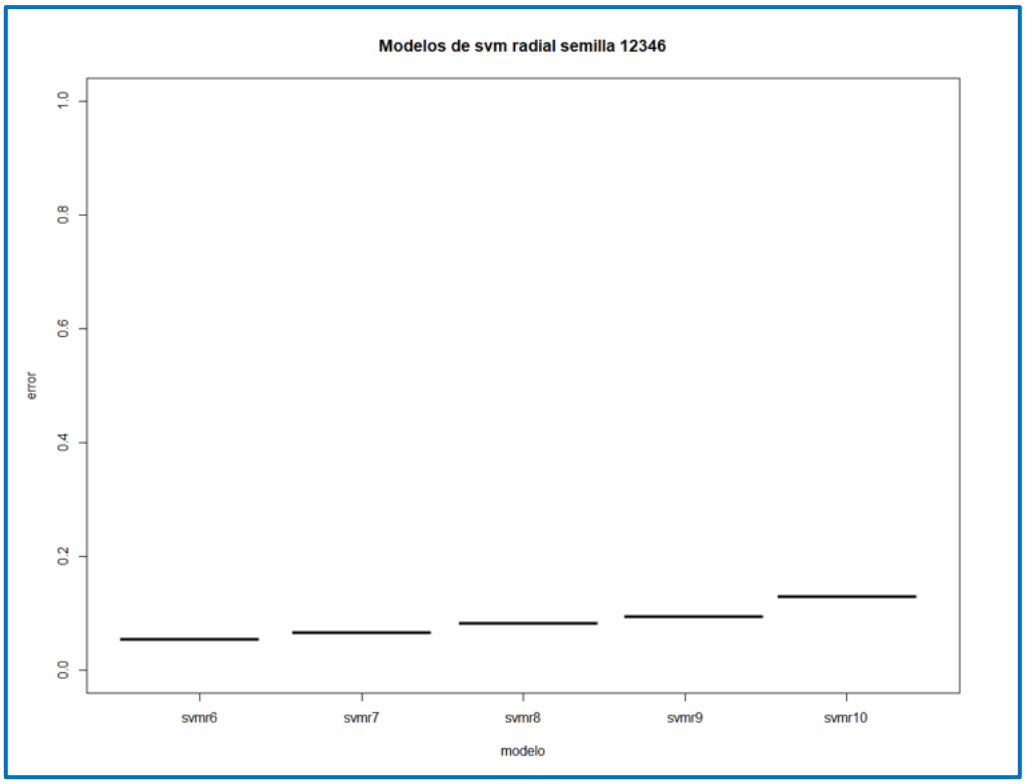


Figura 87: Modelos de SVM con kernel radial semilla 12346

- *Modelos de SVM con kernel lineal y radial semilla 12349*

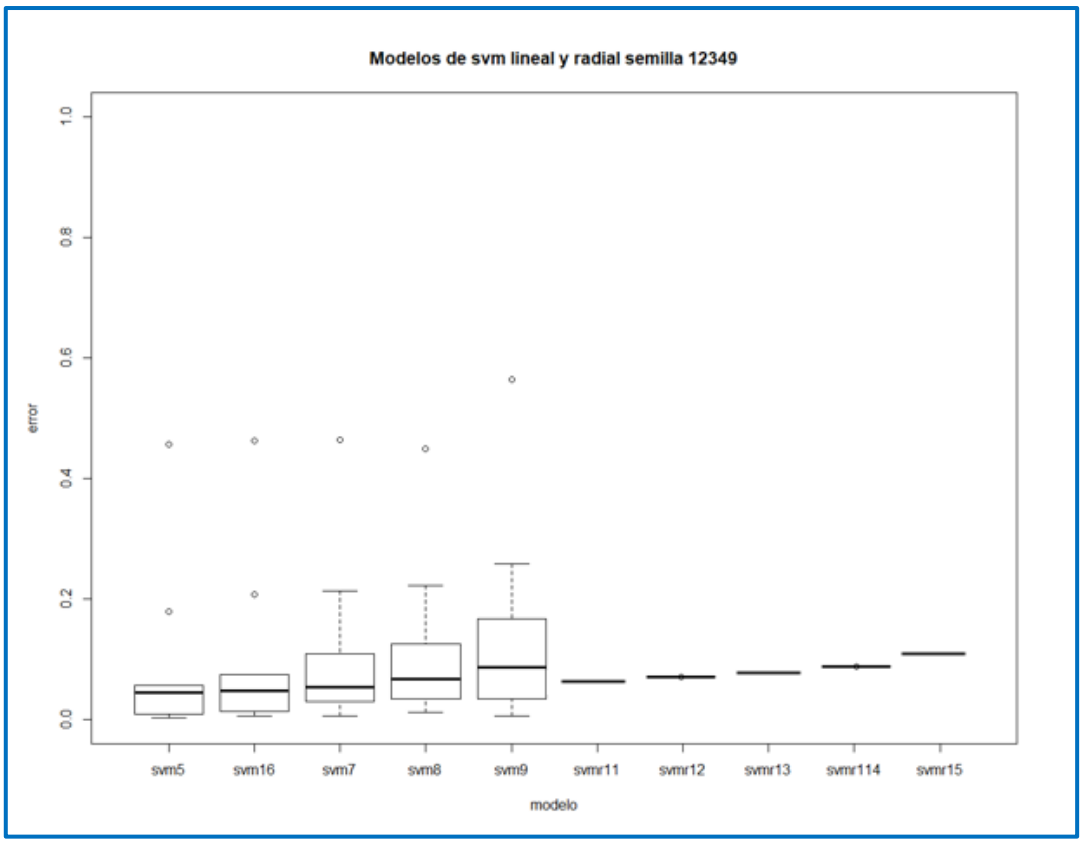


Figura 85: Comparación de modelos de SVM semilla 12349

A continuación, separamos por comportamientos similares, los modelos de la Figura 85 ya que no se aprecian correctamente. Es decir, los modelos $svm - sv4$ formarán parte de un mismo gráfico (Figura 88), mientras que los demás, corresponderán a otro gráfico (Figura 89).,

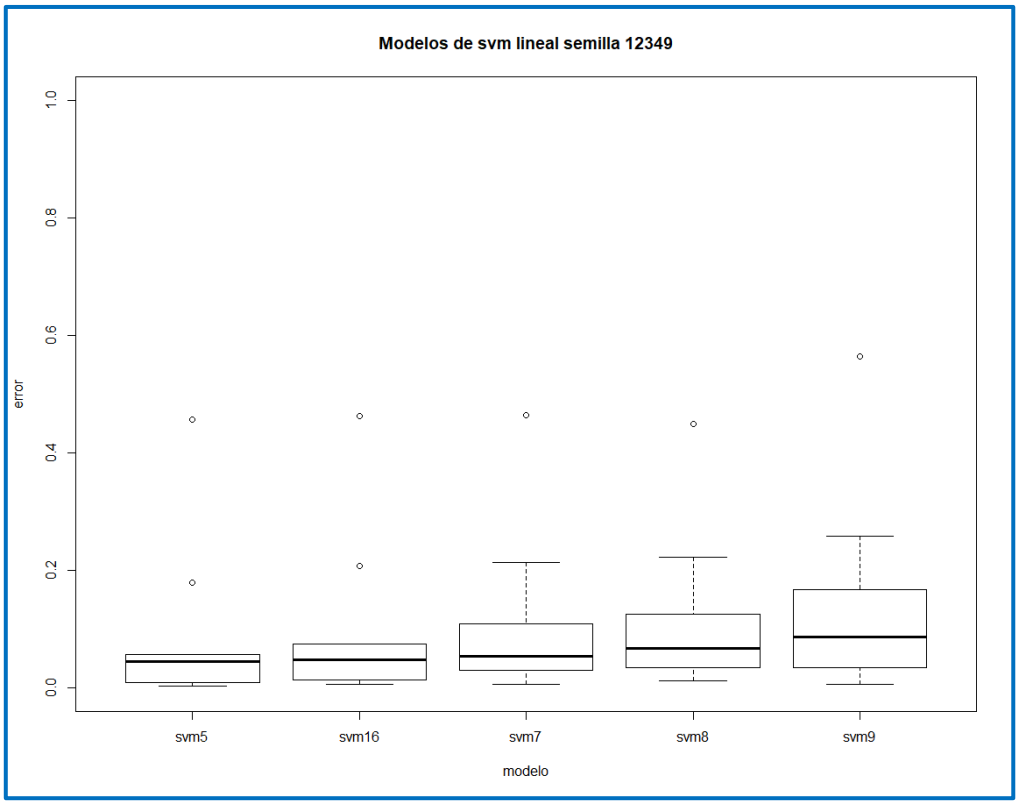


Figura 88: Modelos de SVM con kernel lineal semilla 12349

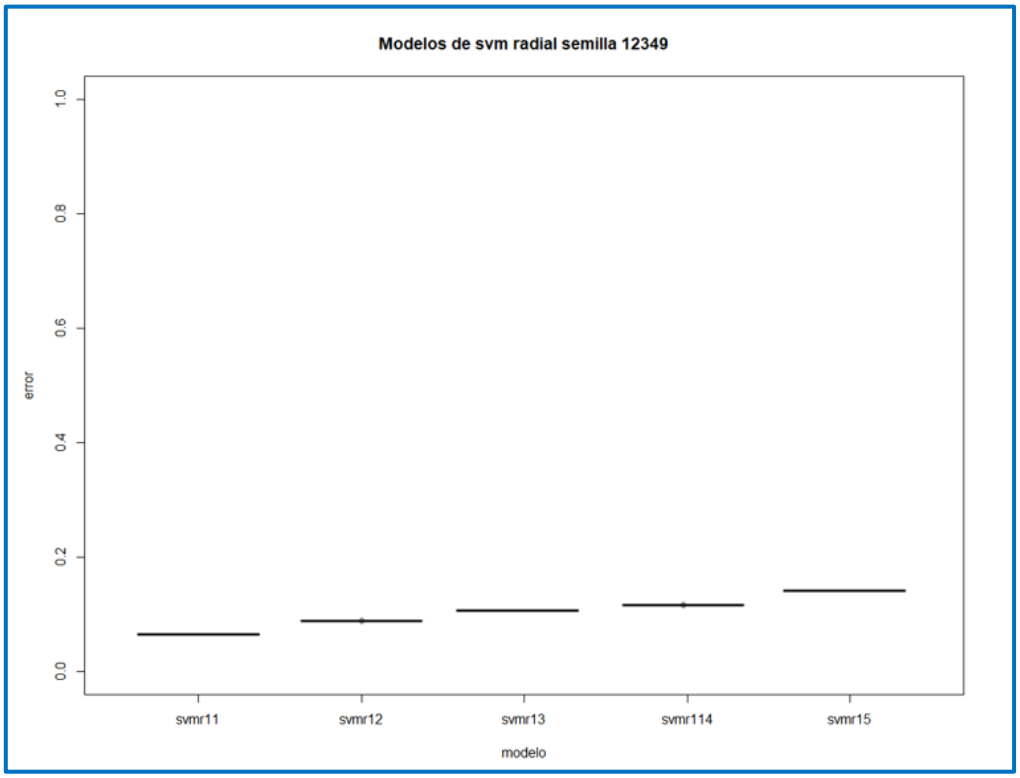


Figura 89: Modelos de SVM con kernel radial semilla 12349

I Código R empleado

```
#####################################################################
#Lectura de datos
#####################################################################
file = "C:/Users/Beatriz/Desktop/TFM_def/precios-gasolina-defin.csv"
precios_gasolDEF <- read.csv(file=file, header = TRUE, sep = ";", encoding="UTF-8")
#encoding="UTF-8" es para que lea los acentos

attach(precios_gasolDEF)

#############################################################
#Visualización de los datos
#############################################################
summary(precios_gasol_def) #nos fijamos que hay duplicidades. Por lo que tenemos que crear un conjunto de datos con las filas no repetidas.

#############################################################
#Creación del conjunto de datos sin las variables variantes, "gasolineras"
#############################################################
columnasInt <- precios_gasol[,c(2:9, 11:13)] #Seleccionamos las colunmas que nos interesan para la creación de este conjunto de datos(sin fecha, sin precio y sin tipo_gasol)
duplicados <- duplicated(columnasInt) #Nos dice qué valores son duplicidades (TRUE)
gasolineras <- columnasInt[!duplicados,] #Creamos el conjunto de datos de las gasolineras (sin la variable precio).
  #Por lo que ahora trabajaremos con este conjunto de datos.

#Creamos un csv del conjunto de datos de las gasolineras
write.table(gasolineras,file="C:/Users/Beatriz/Desktop/TFM_def/gasolineras.csv", sep=";", quote = FALSE, row.names=FALSE, dec = ",")

summary(gasolineras)

#Comprobación de las duplicidades
duplic1 <- which(gasolineras$direccion == "AVENIDA ANDALUCIA, S/N")
View(gasolineras[duplic1,]) #Efectivamente corresponde a las 28 gasolineras

duplic2 <- which(gasolineras$direccion == "AVENIDA JUAN CARLOS I, S/N")
View(gasolineras[duplic2,]) #Efectivamente corresponde a las 18 gasolineras

duplic3 <- which(gasolineras$direccion == "AVENIDA DEL MEDITERRANEO, S/N")
View(gasolineras[duplic3,]) #Efectivamente corresponde a las 11 gasolineras

#############################################################
#Análisis descriptivo de las variables
#precios_gasolDEF (precios_gasol_def) y gasolineras
#############################################################
attach(precios_gasolDEF)
attach(gasolineras)
```

```
library(ggplot2)

#DIRECCIÓN -> no

#CP
levels(gasolineras$cp) #1000 + 3323 = 4323 niveles

#Sacamos solo las 9 gasolineras más frecuentes
CpMasFrecuente = head(sort(table(gasolineras$cp), decreasing=T), 9)
CpMasFrecuente = names(CpMasFrecuente)
CpMasFrecuente_df = gasolineras[is.element(gasolineras$cp, CpMasFrecuente),]
ggplot(CpMasFrecuente_df, aes(x=CpMasFrecuente_df$cp)) +
  geom_bar() + labs(title="Gráfico de los 9 código postales más frecuentes",
                    x ="cp", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#HORARIO
levels(gasolineras$horario) #557 niveles

#Sacamos solo las 20 gasolineras más frecuentes
HorarioMasFrecuente = head(sort(table(gasolineras$horario), decreasing=T), 20)
HorarioMasFrecuente = names(HorarioMasFrecuente)
HorarioMasFrecuente_df = gasolineras[is.element(gasolineras$horario, HorarioMasFrecuente),]
ggplot(HorarioMasFrecuente_df, aes(horario) ) +
  geom_bar() + coord_flip() + labs(title="Gráfico de los 20 horarios más frecuentes",
                                   x ="horario", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#LATITUD
ggplot(gasolineras, aes(latitud)) +
  geom_histogram(color="white") + labs(title="Gráfico de la latitud",
                                       x ="latitud", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#LONGITUD
ggplot(gasolineras, aes(longitud)) +
  geom_histogram(color="white") + labs(title="Gráfico de la longitud",
                                       x ="longitud", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#LOCALIDAD
levels(gasolineras$localidad) #999+2552 = 3551 niveles

#Sacamos solo las 39 gasolineras más frecuentes
LocalidadMasFrecuente = head(sort(table(gasolineras$localidad), decreasing=T), 39)
LocalidadMasFrecuente = names(LocalidadMasFrecuente)
LocalidadMasFrecuente_df = gasolineras[is.element(gasolineras$localidad,
LocalidadMasFrecuente),]
ggplot(LocalidadMasFrecuente_df, aes(localidad) ) +
  geom_bar() + coord_flip() + labs(title="Gráfico de las 39 localidades más frecuentes",
                                   x ="localidad", y = "frecuencia") +
```

```
  theme(plot.title = element_text(hjust = 0.5))

#MARGEN
ggplot(gasolineras, aes(margen) ) +
  geom_bar() + coord_flip() + labs(title="Gráfico del margen",
                           x ="margen", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#MUNICIPIO
levels(gasolineras$municipio) #999+2147 = 3146 niveles

#Sacamos solo las 47 gasolineras más frecuentes
MunicipioMasFrecuente = head(sort(table(gasolineras$municipio), decreasing=T), 47)
MunicipioMasFrecuente = names(MunicipioMasFrecuente)
MunicipioMasFrecuente_df = gasolineras[is.element(gasolineras$municipio,
MunicipioMasFrecuente),]
ggplot(MunicipioMasFrecuente_df, aes(municipio) ) +
  geom_bar() + coord_flip() + labs(title="Gráfico de los 47 municipios más frecuentes",
                           x ="municipio", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#RÓTULO
levels(gasolineras$rotulo) #1000+2543 = 3543 niveles

#Sacamos solo las 20 gasolineras más frecuentes
RotuloMasFrecuente = head(sort(table(gasolineras$rotulo), decreasing=T), 20)
RotuloMasFrecuente = names(RotuloMasFrecuente)
RotuloMasFrecuente_df = gasolineras[is.element(gasolineras$rotulo, RotuloMasFrecuente),]
ggplot(RotuloMasFrecuente_df, aes(x=RotuloMasFrecuente_df$rotulo)) +
  geom_bar() + labs(title="Gráfico de las 20 rótulos más frecuentes",
                x ="rótulo", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#PROVINCIA
ggplot(gasolineras, aes(provincia) ) +
  geom_bar() + coord_flip() + labs(title="Gráfico de las provincias",
                           x ="provincia", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#TIPO_GASOL
precios_gasolDEF[,"media_precio"] =
as.numeric(as.character(precios_gasol[,"media_precio"]))
ggplot(gasolineras, aes(tipo_gasol) ) +
  geom_bar() + coord_flip() + labs(title="Gráfico de los tipos de gasolina",
                           x ="tipo de gasolina", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#PRECIO
jpeg('rplot.jpg', width = 1920, height = 1080)
boxplot(precios_gasol_def$media_precio ~ as.character(precios_gasol_def$tipo_gasol), main =
"Gráfico del precio en función del
```

```
      tipo de gasolina", xlab = "tipo gasolina", ylab = "precio")
dev.off()

#FECHA
as.character(precios_gasolDEF$fecha[1]) #inicio
as.character(precios_gasolDEF$fecha[length(precios_gasolDEF$fecha)]) #fin

#############################################################################
#Variables de la competencia
#############################################################################
#################gasoleoA
#Creamos un conjunto de datos solo para el tipo de gasolina "gasoleoA"
gasolinerasGasoleoA <- which(gasolineras$tipo_gasol == "gasoleoA")
gasolinerasGasoleoA1 <- gasolineras[gasolinerasGasoleoA,] #conjunto de datos a utilizar

M = matrix(0, nrow = nrow(gasolinerasGasoleoA1), ncol = nrow(gasolinerasGasoleoA1))
library("geosphere")

# Matriz de distancias
for (i in 1:nrow(gasolinerasGasoleoA1)){
  for (j in 1:nrow(gasolinerasGasoleoA1)) {
    if (i < j) {
      p1 = c(gasolinerasGasoleoA1[i,"longitud"], gasolinerasGasoleoA1[i,"latitud"])
      p2 = c(gasolinerasGasoleoA1[j,"longitud"], gasolinerasGasoleoA1[j,"latitud"])
      dist = distHaversine(p1, p2) #distancia del semiverseno
      M[i,j] = dist
      M[j,i] = dist
    }
  }
}

#Calculamos la matriz a 5, 10,20, y 50 km
M5A = M <= 5000 #en metros
M10A = M <= 10000 #en metros
M20A = M <= 20000 #en metros
M50A = M <= 50000 #en metros


#################gasolina98
#Creamos un conjunto de datos solo para el tipo de gasolina "gasolina98"
gasolinerasGasolina98 <- which(gasolineras$tipo_gasol == "gasolina98")
gasolinerasGasolina981 <- gasolineras[gasolinerasGasolina98,] #conjunto de datos a utilizar

M1 = matrix(0, nrow = nrow(gasolinerasGasolina981), ncol = nrow(gasolinerasGasolina981))
library("geosphere")

# Matriz de distancias
for (i in 1:nrow(gasolinerasGasolina981)){
  for (j in 1:nrow(gasolinerasGasolina981)) {
    if (i < j) {
      p1 = c(gasolinerasGasolina981[i,"longitud"], gasolinerasGasolina981[i,"latitud"])
```

```
    p2 = c(gasolinerasGasolina981[j,"longitud"], gasolinerasGasolina981[j,"latitud"])
    dist = distHaversine(p1, p2) #distancia del semiverseno
    M1[i,j] = dist
    M1[j,i] = dist
   }
  }
}

#Calculamos la matriz a 5, 10, 20, y 50 km
M5_98 = M1 <= 5000 #en metros
M10_98 = M1 <= 10000 #en metros
M20_98 = M1 <= 20000 #en metros
M50_98 = M1 <= 50000 #en metros

#############################################################################
#Análisis descriptivo de las variables de la competencia
#############################################################################
#################gasoleoA
#Para cada tipo de gasolina y radio vamos a calcular la media y el número de gasolineras que
hay por radio
#M5
for (i in 1:nrow(gasolinerasGasoleoA1)) {
 gasolineras5 = gasolinerasGasoleoA1[M5A[i,],c("direccion", "latitud", "longitud", "localidad",
"margen", "provincia",
                                               "cp", "horario", "municipio", "rotulo")] #seleccionamos las
gasolineras a 5 km

 media_precios = precios_gasol_def[
  which(
   is.element(
    precios_gasol_def[,"direccion"], gasolineras5[,"direccion"]
   ) &
    precios_gasol_def[,"tipo_gasol"] == "gasoleoA" &
    is.element(
     precios_gasol_def[,"latitud"], gasolineras5[,"latitud"]
    ) &
    is.element(
     precios_gasol_def[,"longitud"], gasolineras5[,"longitud"]
    ) &
    is.element(
     precios_gasol_def[,"localidad"], gasolineras5[,"localidad"]) &
    is.element(
     precios_gasol_def[,"margen"], gasolineras5[,"margen"]
    ) &
    is.element(
     precios_gasol_def[,"provincia"], gasolineras5[,"provincia"]
    ) &
    is.element(
     precios_gasol_def[,"cp"], gasolineras5[,"cp"]
    ) &
    is.element(
```

```
      precios_gasol_def[,"horario"], gasolineras5[,"horario"]
    ) &
    is.element(
      precios_gasol_def[,"municipio"], gasolineras5[,"municipio"]
    ) &
    is.element(
      precios_gasol_def[,"rotulo"], gasolineras5[,"rotulo"]
    )), c("media_precio")] #seleccionamos el precio medio de cada una de las gasolineras
seleccionadas con competencia a 5 km

  gasolinerasGasoleoA1[i, "media_precio_cluster"] = mean(media_precios) #media del precio
de cada gasolinera
  gasolinerasGasoleoA1[i, "numero_gasolineras_cluster"] = dim(gasolineras5)[1] #número de
gasolineras por estación contando ella misma
  gasolinerasGasoleoA1[i, "mínimo_precio_cluster"] = min(media_precios) #mínimo del precio
de cada gasolinera

}

mean(gasolinerasGasoleoA1[,12], na.rm = TRUE) #media total de todas las gasolineras a 5 km
es 1.131645
numeroGasolinerasGasoleoA5km <- gasolinerasGasoleoA1[,13]-1
ggplot(gasolinerasGasoleoA1, aes(numeroGasolinerasGasoleoA5km)) +
  geom_histogram(color="white") + labs(title="Histograma del número de gasolineras a 5 km",
                                       x ="número de gasolineras", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

preciosGasoleoA5km <- gasolinerasGasoleoA1[,12]

#Creamos un nuevo conjunto de datos con las variables de competencia de las medias
precios_competenciaGasoleoA = data.frame(preciosGasoleoA5km)
colnames(precios_competenciaGasoleoA) <- c("precio_5km")
precios_competenciaGasoleoA <-
as.data.frame(precios_competenciaGasoleoA[complete.cases(precios_competenciaGasoleoA),
])
colnames(precios_competenciaGasoleoA) <- c("precio_5km")

mean(gasolinerasGasoleoA1[,14], na.rm = TRUE) #media del mínimo de todas las gasolineras a
5 km es 1.049887
min_preciosGasoleoA5km <- gasolinerasGasoleoA1[,14]

#Creamos un nuevo conjunto de datos con las variables de competencia del mínimo
min_precios_competenciaGasoleoA = data.frame(min_preciosGasoleoA5km)
colnames(min_precios_competenciaGasoleoA) <- c("precio_5km")
precios_competenciaGasoleoA <-
as.data.frame(min_precios_competenciaGasoleoA[complete.cases(min_precios_competenciaGa
soleoA), ])
colnames(min_precios_competenciaGasoleoA) <- c("precio_5km")

#M10
for (i in 1:nrow(gasolinerasGasoleoA1)) {
```

```
  gasolineras10 = gasolinerasGasoleoA1[M10A[i,],c("direccion", "latitud", "longitud",
"localidad", "margen", "provincia",
                                    "cp", "horario", "municipio", "rotulo")]

  media_precios = precios_gasol_def[
    which(
      is.element(
        precios_gasol_def[,"direccion"], gasolineras10[,"direccion"]
      ) &
        precios_gasol_def[,"tipo_gasol"] == "gasoleoA" &
        is.element(
          precios_gasol_def[,"latitud"], gasolineras10[,"latitud"]
        ) &
        is.element(
          precios_gasol_def[,"longitud"], gasolineras10[,"longitud"]
        ) &
        is.element(
          precios_gasol_def[,"localidad"], gasolineras10[,"localidad"]) &
        is.element(
          precios_gasol_def[,"margen"], gasolineras10[,"margen"]
        ) &
        is.element(
          precios_gasol_def[,"provincia"], gasolineras10[,"provincia"]
        ) &
        is.element(
          precios_gasol_def[,"cp"], gasolineras10[,"cp"]
        ) &
        is.element(
          precios_gasol_def[,"horario"], gasolineras10[,"horario"]
        ) &
        is.element(
          precios_gasol_def[,"municipio"], gasolineras10[,"municipio"]
        ) &
        is.element(
          precios_gasol_def[,"rotulo"], gasolineras10[,"rotulo"]
        )), c("media_precio")]

  gasolinerasGasoleoA1[i, "media_precio_cluster"] = mean(media_precios)
  gasolinerasGasoleoA1[i, "numero_gasolineras_cluster"] = dim(gasolineras10)[1]
  gasolinerasGasoleoA1[i, "mínimo_precio_cluster"] = min(media_precios)

}

mean(gasolinerasGasoleoA1[,12], na.rm = TRUE) #media total de todas las gasolineras a 10 km
es 1.131787
numeroGasolinerasGasoleoA10km <- gasolinerasGasoleoA1[,13]-1
ggplot(gasolinerasGasoleoA1, aes(numeroGasolinerasGasoleoA10km)) +
  geom_histogram(color="white") + labs(title="Histograma del número de gasolineras a 10 km",
                                       x ="número de gasolineras", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
preciosGasoleoA10km <- gasolinerasGasoleoA1[,12]

#Añadimos al conjunto de datos la siguiente variable de precio con las variables de competencia
precios_competenciaGasoleoA = data.frame(preciosGasoleoA5km, preciosGasoleoA10km)
colnames(precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km")
precios_competenciaGasoleoA <-
as.data.frame(precios_competenciaGasoleoA[complete.cases(precios_competenciaGasoleoA),
])
colnames(precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km")

mean(gasolinerasGasoleoA1[,14], na.rm = TRUE) #media del mínimo de todas las gasolineras a
10 km es 1.030727
min_preciosGasoleoA10km <- gasolinerasGasoleoA1[,14]

#Creamos un nuevo conjunto de datos con las variables de competencia del mínimo
min_precios_competenciaGasoleoA = data.frame(min_preciosGasoleoA5km,
min_preciosGasoleoA10km)
colnames(min_precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km")
precios_competenciaGasoleoA <-
as.data.frame(min_precios_competenciaGasoleoA[complete.cases(min_precios_competenciaGa
soleoA), ])
colnames(min_precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km")

#M20
for (i in 1:nrow(gasolinerasGasoleoA1)) {
  gasolineras20 = gasolinerasGasoleoA1[M20A[i,],c("direccion", "latitud", "longitud",
"localidad", "margen", "provincia",
                                        "cp", "horario", "municipio", "rotulo")]

  media_precios = precios_gasol_def[
    which(
      is.element(
        precios_gasol_def[,"direccion"], gasolineras20[,"direccion"]
      ) &
        precios_gasol_def[,"tipo_gasol"] == "gasoleoA" &
        is.element(
          precios_gasol_def[,"latitud"], gasolineras20[,"latitud"]
        ) &
        is.element(
          precios_gasol_def[,"longitud"], gasolineras20[,"longitud"]
        ) &
        is.element(
          precios_gasol_def[,"localidad"], gasolineras20[,"localidad"]) &
        is.element(
          precios_gasol_def[,"margen"], gasolineras20[,"margen"]
        ) &
        is.element(
          precios_gasol_def[,"provincia"], gasolineras20[,"provincia"]
        ) &
        is.element(
          precios_gasol_def[,"cp"], gasolineras20[,"cp"]
```

```
      ) &
      is.element(
        precios_gasol_def[,"horario"], gasolineras20[,"horario"]
      ) &
      is.element(
        precios_gasol_def[,"municipio"], gasolineras20[,"municipio"]
      ) &
      is.element(
        precios_gasol_def[,"rotulo"], gasolineras20[,"rotulo"]
      )), c("media_precio")]

  gasolinerasGasoleoA1[i, "media_precio_cluster"] = mean(media_precios)
  gasolinerasGasoleoA1[i, "numero_gasolineras_cluster"] = dim(gasolineras20)[1]
  gasolinerasGasoleoA1[i, "mínimo_precio_cluster"] = min(media_precios)

}

mean(gasolinerasGasoleoA1[,12], na.rm = TRUE) #media total de todas las gasolineras a 20 km
es 1.132142
numeroGasolinerasGasoleoA20km <- gasolinerasGasoleoA1[,13]-1
ggplot(gasolinerasGasoleoA1, aes(numeroGasolinerasGasoleoA20km)) +
  geom_histogram(color="white") + labs(title="Histograma del número de gasolineras a 20 km",
                                       x ="número de gasolineras", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

preciosGasoleoA20km <- gasolinerasGasoleoA1[,12]

#Añadimos al conjunto de datos la siguiente variable de precio con las variables de competencia
precios_competenciaGasoleoA = data.frame(preciosGasoleoA5km, preciosGasoleoA10km,
preciosGasoleoA20km)
colnames(precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km", "precio_20km")
precios_competenciaGasoleoA <-
as.data.frame(precios_competenciaGasoleoA[complete.cases(precios_competenciaGasoleoA),
])
colnames(precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km", "precio_20km")

mean(gasolinerasGasoleoA1[,14], na.rm = TRUE) #media del mínimo de todas las gasolineras a
20 km es 1.008861
min_preciosGasoleoA20km <- gasolinerasGasoleoA1[,14]

#Creamos un nuevo conjunto de datos con las variables de competencia del mínimo
min_precios_competenciaGasoleoA = data.frame(min_preciosGasoleoA5km,
min_preciosGasoleoA10km, min_preciosGasoleoA20km)
colnames(min_precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km",
"precio_20km")
precios_competenciaGasoleoA <-
as.data.frame(min_precios_competenciaGasoleoA[complete.cases(min_precios_competenciaGa
soleoA), ])
colnames(min_precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km",
"precio_20km")
```

```
#M50
for (i in 1:nrow(gasolinerasGasoleoA1)) {
  gasolineras50 = gasolinerasGasoleoA1[M50A[i,],c("direccion", "latitud", "longitud",
"localidad", "margen", "provincia",
                                        "cp", "horario", "municipio", "rotulo")]

  media_precios = precios_gasol_def[
    which(
      is.element(
        precios_gasol_def[,"direccion"], gasolineras50[,"direccion"]
      ) &
        precios_gasol_def[,"tipo_gasol"] == "gasoleoA" &
        is.element(
          precios_gasol_def[,"latitud"], gasolineras50[,"latitud"]
        ) &
        is.element(
          precios_gasol_def[,"longitud"], gasolineras50[,"longitud"]
        ) &
        is.element(
          precios_gasol_def[,"localidad"], gasolineras50[,"localidad"]) &
        is.element(
          precios_gasol_def[,"margen"], gasolineras50[,"margen"]
        ) &
        is.element(
          precios_gasol_def[,"provincia"], gasolineras50[,"provincia"]
        ) &
        is.element(
          precios_gasol_def[,"cp"], gasolineras50[,"cp"]
        ) &
        is.element(
          precios_gasol_def[,"horario"], gasolineras50[,"horario"]
        ) &
        is.element(
          precios_gasol_def[,"municipio"], gasolineras50[,"municipio"]
        ) &
        is.element(
          precios_gasol_def[,"rotulo"], gasolineras50[,"rotulo"]
        )), c("media_precio")]

  gasolinerasGasoleoA1[i, "media_precio_cluster"] = mean(media_precios)
  gasolinerasGasoleoA1[i, "numero_gasolineras_cluster"] = dim(gasolineras50)[1]
  gasolinerasGasoleoA1[i, "mínimo_precio_cluster"] = min(media_precios)

}

mean(gasolinerasGasoleoA1[,12], na.rm = TRUE) #media total de todas las gasolineras a 50 km
es 1.131882
numeroGasolinerasGasoleoA50km <- gasolinerasGasoleoA1[,13]-1
ggplot(gasolinerasGasoleoA1, aes(numeroGasolinerasGasoleoA50km)) +
  geom_histogram(color="white") + labs(title="Histograma del número de gasolineras a 50 km",
                                       x ="número de gasolineras", y = "frecuencia") +
```

```
  theme(plot.title = element_text(hjust = 0.5))

preciosGasoleoA50km <- gasolinerasGasoleoA1[,12]

#Añadimos al conjunto de datos la siguiente variable de precio con las variables de competencia
precios_competenciaGasoleoA = data.frame(preciosGasoleoA5km, preciosGasoleoA10km,
preciosGasoleoA20km, preciosGasoleoA50km)
colnames(precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km", "precio_20km",
"precio_50km")
precios_competenciaGasoleoA <-
as.data.frame(precios_competenciaGasoleoA[complete.cases(precios_competenciaGasoleoA),
])
colnames(precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km", "precio_20km",
"precio_50km")

mean(gasolinerasGasoleoA1[,14], na.rm = TRUE) #media del mínimo de todas las gasolineras a
50 km es 0.984272
min_preciosGasoleoA50km <- gasolinerasGasoleoA1[,14]

#Creamos un nuevo conjunto de datos con las variables de competencia del mínimo
min_precios_competenciaGasoleoA = data.frame(min_preciosGasoleoA5km,
min_preciosGasoleoA10km, min_preciosGasoleoA20km, min_preciosGasoleoA50km)
colnames(min_precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km",
"precio_20km", "precio_50km")
precios_competenciaGasoleoA <-
as.data.frame(min_precios_competenciaGasoleoA[complete.cases(min_precios_competenciaGa
soleoA), ])
colnames(min_precios_competenciaGasoleoA) <- c("precio_5km", "precio_10km",
"precio_20km", "precio_50km")


##################gasolina98
#M5
for (i in 1:nrow(gasolinerasGasolina981)) {
  gasolineras5_98 = gasolinerasGasolina981[M5_98[i,],c("direccion", "latitud", "longitud",
"localidad", "margen", "provincia",
                                    "cp", "horario", "municipio", "rotulo")] #seleccionamos las
gasolineras a 5 km

  media_precios = precios_gasol_def[
    which(
      is.element(
        precios_gasol_def[,"direccion"], gasolineras5_98[,"direccion"]
      ) &
        precios_gasol_def[,"tipo_gasol"] == "gasoleoA" &
        is.element(
          precios_gasol_def[,"latitud"], gasolineras5_98[,"latitud"]
        ) &
        is.element(
          precios_gasol_def[,"longitud"], gasolineras5_98[,"longitud"]
        ) &
```

```
      is.element(
        precios_gasol_def[,"localidad"], gasolineras5_98[,"localidad"]) &
      is.element(
        precios_gasol_def[,"margen"], gasolineras5_98[,"margen"]
      ) &
      is.element(
        precios_gasol_def[,"provincia"], gasolineras5_98[,"provincia"]
      ) &
      is.element(
        precios_gasol_def[,"cp"], gasolineras5_98[,"cp"]
      ) &
      is.element(
        precios_gasol_def[,"horario"], gasolineras5_98[,"horario"]
      ) &
      is.element(
        precios_gasol_def[,"municipio"], gasolineras5_98[,"municipio"]
      ) &
      is.element(
        precios_gasol_def[,"rotulo"], gasolineras5_98[,"rotulo"]
      )), c("media_precio")] #seleccionamos el precio medio de cada una de las gasolineras
seleccionadas con competencia a 5 km

  gasolinerasGasolina981[i, "media_precio_cluster"] = mean(media_precios) #media del precio
de cada gasolinera
  gasolinerasGasolina981[i, "numero_gasolineras_cluster"] = dim(gasolineras5_98)[1] #número
de gasolineras por estación contando ella misma
  gasolinerasGasolina981[i, "mínimo_precio_cluster"] = min(media_precios) #mínimo del
precio de cada gasolinera
}

mean(gasolinerasGasolina981[,12], na.rm = TRUE) #media total de todas las gasolineras a 5 km
es 1.146103
numeroGasolinerasGasolina981_5km <- gasolinerasGasolina981[,13]-1
ggplot(gasolinerasGasolina981, aes(numeroGasolinerasGasolina981_5km)) +
  geom_histogram(color="white") + labs(title="Histograma del número de gasolineras a 5 km",
                                       x ="número de gasolineras", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

preciosGasolina98_5km <- gasolinerasGasolina981[,12]

#Creamos un nuevo conjunto de datos con las variables de competencia
precios_competenciaGasolina98 = data.frame(preciosGasolina98_5km)
colnames(precios_competenciaGasolina98) <- c("precio_5km")
precios_competenciaGasolina98 <-
as.data.frame(precios_competenciaGasolina98[complete.cases(precios_competenciaGasolina98
), ])
colnames(precios_competenciaGasolina98) <- c("precio_5km")

mean(gasolinerasGasolina981[,14], na.rm = TRUE) #media del mínimo de todas las gasolineras
a 5 km es 1.076075
min_preciosGasolina98_5km <- gasolinerasGasolina981[,14]
```

```
#Creamos un nuevo conjunto de datos con las variables de competencia del mínimo
min_precios_competenciaGasolina98 = data.frame(min_preciosGasolina98_5km)
colnames(min_precios_competenciaGasolina98) <- c("precio_5km")
precios_competenciaGasolina98 <-
as.data.frame(min_precios_competenciaGasolina98[complete.cases(min_precios_competenciaG
asolina98), ])
colnames(min_precios_competenciaGasolina98) <- c("precio_5km")

#M10
for (i in 1:nrow(gasolinerasGasolina981)) {
  gasolineras10_98 = gasolinerasGasolina981[M10_98[i,],c("direccion", "latitud", "longitud",
"localidad", "margen", "provincia",
                                        "cp", "horario", "municipio", "rotulo")]

  media_precios = precios_gasol_def[
    which(
      is.element(
        precios_gasol_def[,"direccion"], gasolineras10_98[,"direccion"]
      ) &
        precios_gasol_def[,"tipo_gasol"] == "gasoleoA" &
        is.element(
          precios_gasol_def[,"latitud"], gasolineras10_98[,"latitud"]
        ) &
        is.element(
          precios_gasol_def[,"longitud"], gasolineras10_98[,"longitud"]
        ) &
        is.element(
          precios_gasol_def[,"localidad"], gasolineras10_98[,"localidad"]) &
        is.element(
          precios_gasol_def[,"margen"], gasolineras10_98[,"margen"]
        ) &
        is.element(
          precios_gasol_def[,"provincia"], gasolineras10_98[,"provincia"]
        ) &
        is.element(
          precios_gasol_def[,"cp"], gasolineras10_98[,"cp"]
        ) &
        is.element(
          precios_gasol_def[,"horario"], gasolineras10_98[,"horario"]
        ) &
        is.element(
          precios_gasol_def[,"municipio"], gasolineras10_98[,"municipio"]
        ) &
        is.element(
          precios_gasol_def[,"rotulo"], gasolineras10_98[,"rotulo"]
        )), c("media_precio")]

  gasolinerasGasolina981[i, "media_precio_cluster"] = mean(media_precios)
  gasolinerasGasolina981[i, "numero_gasolineras_cluster"] = dim(gasolineras10_98)[1]
  gasolinerasGasolina981[i, "mínimo_precio_cluster"] = min(media_precios)
```

```
}

mean(gasolinerasGasolina981[,12], na.rm = TRUE) #media total de todas las gasolineras a 10
km es 1.146226
numeroGasolinerasGasolina981_10km <- gasolinerasGasolina981[,13]-1
ggplot(gasolinerasGasolina981, aes(numeroGasolinerasGasolina981_10km)) +
  geom_histogram(color="white") + labs(title="Histograma del número de gasolineras a 10 km",
                                       x ="número de gasolineras", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

preciosGasolina98_10km <- gasolinerasGasolina981[,12]

#Añadimos al conjunto de datos la siguiente variable de precio con las variables de competencia
precios_competenciaGasolina98 = data.frame(preciosGasolina98_5km,
preciosGasolina98_10km)
colnames(precios_competenciaGasolina98) <- c("precio_5km", "precio_10km")
precios_competenciaGasolina98 <-
as.data.frame(precios_competenciaGasolina98[complete.cases(precios_competenciaGasolina98
), ])
colnames(precios_competenciaGasolina98) <- c("precio_5km", "precio_10km")

mean(gasolinerasGasolina981[,14], na.rm = TRUE) #media del mínimo de todas las gasolineras
a 10 km es 1.057459
min_preciosGasolina98_10km <- gasolinerasGasolina981[,14]

#Creamos un nuevo conjunto de datos con las variables de competencia del mínimo
min_precios_competenciaGasolina98 = data.frame(min_preciosGasolina98_5km,
min_preciosGasolina98_10km)
colnames(min_precios_competenciaGasolina98) <- c("precio_5km", "precio_10km")
precios_competenciaGasolina98 <-
as.data.frame(min_precios_competenciaGasolina98[complete.cases(min_precios_competenciaG
asolina98), ])
colnames(min_precios_competenciaGasolina98) <- c("precio_5km", "precio_10km")

#M20
for (i in 1:nrow(gasolinerasGasolina981)) {
  gasolineras20_98 = gasolinerasGasolina981[M20_98[i,],c("direccion", "latitud", "longitud",
"localidad", "margen", "provincia",
                                       "cp", "horario", "municipio", "rotulo")]

  media_precios = precios_gasol_def[
    which(
      is.element(
        precios_gasol_def[,"direccion"], gasolineras20_98[,"direccion"]
      ) &
        precios_gasol_def[,"tipo_gasol"] == "gasoleoA" &
        is.element(
          precios_gasol_def[,"latitud"], gasolineras20_98[,"latitud"]
        ) &
        is.element(
          precios_gasol_def[,"longitud"], gasolineras20_98[,"longitud"]
```

```
      ) &
      is.element(
        precios_gasol_def[,"localidad"], gasolineras20_98[,"localidad"]) &
      is.element(
        precios_gasol_def[,"margen"], gasolineras20_98[,"margen"]
      ) &
      is.element(
        precios_gasol_def[,"provincia"], gasolineras20_98[,"provincia"]
      ) &
      is.element(
        precios_gasol_def[,"cp"], gasolineras20_98[,"cp"]
      ) &
      is.element(
        precios_gasol_def[,"horario"], gasolineras20_98[,"horario"]
      ) &
      is.element(
        precios_gasol_def[,"municipio"], gasolineras20_98[,"municipio"]
      ) &
      is.element(
        precios_gasol_def[,"rotulo"], gasolineras20_98[,"rotulo"]
      )), c("media_precio")]

  gasolinerasGasolina981[i, "media_precio_cluster"] = mean(media_precios)
  gasolinerasGasolina981[i, "numero_gasolineras_cluster"] = dim(gasolineras20_98)[1]
  gasolinerasGasolina981[i, "mínimo_precio_cluster"] = min(media_precios)
}

mean(gasolinerasGasolina981[,12], na.rm = TRUE) #media total de todas las gasolineras a 20
km es 1.146358
numeroGasolinerasGasolina981_20km <- gasolinerasGasolina981[,13]-1
ggplot(gasolinerasGasolina981, aes(numeroGasolinerasGasolina981_20km)) +
  geom_histogram(color="white") + labs(title="Histograma del número de gasolineras a 20 km",
                                       x ="número de gasolineras", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

preciosGasolina98_20km <- gasolinerasGasolina981[,12]

#Añadimos al conjunto de datos la siguiente variable de precio con las variables de competencia
precios_competenciaGasolina98 = data.frame(preciosGasolina98_5km,
preciosGasolina98_10km, preciosGasolina98_20km)
colnames(precios_competenciaGasolina98) <- c("precio_5km", "precio_10km",
"precio_20km")
precios_competenciaGasolina98 <-
as.data.frame(precios_competenciaGasolina98[complete.cases(precios_competenciaGasolina98
), ])
colnames(precios_competenciaGasolina98) <- c("precio_5km", "precio_10km",
"precio_20km")

mean(gasolinerasGasolina981[,14], na.rm = TRUE) #media del mínimo de todas las gasolineras
a 20 km es 1.035816
min_preciosGasolina98_20km <- gasolinerasGasolina981[,14]
```

```
#Creamos un nuevo conjunto de datos con las variables de competencia del mínimo
min_precios_competenciaGasolina98 = data.frame(min_preciosGasolina98_5km,
min_preciosGasolina98_10km, min_preciosGasolina98_20km)
colnames(min_precios_competenciaGasolina98) <- c("precio_5km", "precio_10km",
"precio_20km")
precios_competenciaGasolina98 <-
as.data.frame(min_precios_competenciaGasolina98[complete.cases(min_precios_competenciaG
asolina98), ])
colnames(min_precios_competenciaGasolina98) <- c("precio_5km", "precio_10km",
"precio_20km")

#M50
for (i in 1:nrow(gasolinerasGasolina981)) {
  gasolineras50_98 = gasolinerasGasolina981[M50_98[i,],c("direccion", "latitud", "longitud",
"localidad", "margen", "provincia",
                                        "cp", "horario", "municipio", "rotulo")]

  media_precios = precios_gasol_def[
    which(
      is.element(
        precios_gasol_def[,"direccion"], gasolineras50_98[,"direccion"]
      ) &
        precios_gasol_def[,"tipo_gasol"] == "gasoleoA" &
        is.element(
          precios_gasol_def[,"latitud"], gasolineras50_98[,"latitud"]
        ) &
        is.element(
          precios_gasol_def[,"longitud"], gasolineras50_98[,"longitud"]
        ) &
        is.element(
          precios_gasol_def[,"localidad"], gasolineras50_98[,"localidad"]) &
        is.element(
          precios_gasol_def[,"margen"], gasolineras50_98[,"margen"]
        ) &
        is.element(
          precios_gasol_def[,"provincia"], gasolineras50_98[,"provincia"]
        ) &
        is.element(
          precios_gasol_def[,"cp"], gasolineras50_98[,"cp"]
        ) &
        is.element(
          precios_gasol_def[,"horario"], gasolineras50_98[,"horario"]
        ) &
        is.element(
          precios_gasol_def[,"municipio"], gasolineras50_98[,"municipio"]
        ) &
        is.element(
          precios_gasol_def[,"rotulo"], gasolineras50_98[,"rotulo"]
        )), c("media_precio")]

  gasolinerasGasolina981[i, "media_precio_cluster"] = mean(media_precios)
```

```
  gasolinerasGasolina981[i, "numero_gasolineras_cluster"] = dim(gasolineras50_98)[1]
  gasolinerasGasolina981[i, "mínimo_precio_cluster"] = min(media_precios)
}

mean(gasolinerasGasolina981[,12], na.rm = TRUE) #media total de todas las gasolineras a 50
km es 1.146281
numeroGasolinerasGasolina981_50km <- gasolinerasGasolina981[,13]-1
ggplot(gasolinerasGasolina981, aes(numeroGasolinerasGasolina981_50km)) +
  geom_histogram(color="white") + labs(title="Histograma del número de gasolineras a 50 km",
                                       x ="número de gasolineras", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

preciosGasolina98_50km <- gasolinerasGasolina981[,12]

#Añadimos al conjunto de datos la siguiente variable de precio con las variables de competencia
precios_competenciaGasolina98 = data.frame(preciosGasolina98_5km,
preciosGasolina98_10km, preciosGasolina98_20km, preciosGasolina98_50km)
colnames(precios_competenciaGasolina98) <- c("precio_5km", "precio_10km", "precio_20km",
"precio_50km")
precios_competenciaGasolina98 <-
as.data.frame(precios_competenciaGasolina98[complete.cases(precios_competenciaGasolina98
), ])
colnames(precios_competenciaGasolina98) <- c("precio_5km", "precio_10km", "precio_20km",
"precio_50km")

mean(gasolinerasGasolina981[,14], na.rm = TRUE) #media del mínimo de todas las gasolineras
a 50 km es 1.006051
min_preciosGasolina98_50km <- gasolinerasGasolina981[,14]

#Creamos un nuevo conjunto de datos con las variables de competencia del mínimo
min_precios_competenciaGasolina98 = data.frame(min_preciosGasolina98_5km,
min_preciosGasolina98_10km, min_preciosGasolina98_20km,
                                       min_preciosGasolina98_50km)
colnames(min_precios_competenciaGasolina98) <- c("precio_5km", "precio_10km",
"precio_20km", "precio_50km")
precios_competenciaGasolina98 <-
as.data.frame(min_precios_competenciaGasolina98[complete.cases(min_precios_competenciaG
asolina98), ])
colnames(min_precios_competenciaGasolina98) <- c("precio_5km", "precio_10km",
"precio_20km", "precio_50km")

##############################################################################
#Recodificación de las categorías de alguna de las variables
##############################################################################
#PRECIOS_GASOL_DEF
#HORARIO
#Eliminamos los : de alguna de las categorías de la variable horario ya que la función "recode"
#sino no lo reconoce (reemplazar):
precios_gasol_def$horario = as.character(precios_gasol_def$horario)
precios_gasol_def$horario[which(precios_gasol_def$horario == "L-D: 24H")] = "L-D 24H"
precios_gasol_def$horario[which(precios_gasol_def$horario == "L: 24H")] = "L 24H"
```

```
precios_gasol_def$horario[which(precios_gasol_def$horario == "L-V: 24H")] = "L-V 24H"
precios_gasol_def$horario[which(precios_gasol_def$horario == "S-D: 24H")] = "S-D 24H"

library(car)
precios_gasol_def$horario = recode(precios_gasol_def$horario, "c('L-D 24H', 'L 24H', 'L-V
24H', 'S-D 24H') = 'abierto las 24 horas';else = 'no abre las 24 horas'")


#GASOLINERAS
#HORARIO
#Eliminamos los : de alguna de las categorías de la variable horario ya que la función "recode"
#sino no lo reconoce (reemplazar):
gasolineras$horario = as.character(gasolineras$horario)
gasolineras$horario[which(gasolineras$horario == "L-D: 24H")] = "L-D 24H"
gasolineras$horario[which(gasolineras$horario == "L: 24H")] = "L 24H"
gasolineras$horario[which(gasolineras$horario == "L-V: 24H")] = "L-V 24H"
gasolineras$horario[which(gasolineras$horario == "S-D: 24H")] = "S-D 24H"

library(car)
gasolineras$horario = recode(gasolineras$horario, "c('L-D 24H', 'L 24H', 'L-V 24H', 'S-D 24H')
= 'abierto las 24 horas';else = 'no abre las 24 horas'")


#PRECIOS_GASOL_DEF
#ROTULO
#Sacamos solo las 20 gasolineras más frecuentes
RotuloMasFrecuente1 = head(sort(table(gasolineras$rotulo), decreasing=T), 30)
RotuloMasFrecuente1

supermercados = c( "CARREFOUR", "ALCAMPO", "EROSKI", "SIMPLY", "ALCAMPO
S.A.")
primeras_marcas = c( "REPSOL", "CEPSA", "GALP", "SIMPLY", "SHELL", "BP",
"PETRONOR", "CAMPSA")
precios_gasol_def$rotulo = as.character(precios_gasol_def$rotulo)
precios_gasol_def$rotulo[which(is.element(precios_gasol_def$rotulo, supermercados))] =
"supermercado"
precios_gasol_def$rotulo[which(!is.element(precios_gasol_def$rotulo, union(supermercados,
primeras_marcas)))] = "otros"
precios_gasol_def$rotulo = as.factor(precios_gasol_def$rotulo)

#GSOLINERAS
#ROTULO
supermercados1 = c( "CARREFOUR", "ALCAMPO", "EROSKI", "SIMPLY", "ALCAMPO
S.A.")
primeras_marcas1 = c( "REPSOL", "CEPSA", "GALP", "SIMPLY", "SHELL", "BP",
"PETRONOR", "CAMPSA")
gasolineras$rotulo = as.character(gasolineras$rotulo)
gasolineras$rotulo[which(is.element(gasolineras$rotulo, supermercados1))] = "supermercado"
gasolineras$rotulo[which(!is.element(gasolineras$rotulo, union(supermercados1,
primeras_marcas1)))] = "otros"
gasolineras$rotulo = as.factor(gasolineras$rotulo)
```

```
#############################################################################
#Depuración de datos
#############################################################################
###Tratamiento de datos faltantes o missing
which(is.na(precios_gasol_def$fecha))
which(is.na(gasolineras$cp))
which(is.na(gasolineras$horario))
which(is.na(gasolineras$latitud))
which(is.na(gasolineras$longitud))
which(is.na(gasolineras$localidad))
which(is.na(gasolineras$margen))
which(is.na(gasolineras$municipio))
which(is.na(gasolineras$rotulo))
which(is.na(gasolineras$provincia))
which(is.na(precios_gasol_def$tipo_gasol))
which(is.na(precios_gasol_def$media_precio))
which(is.na(gasolineras$direccion))
   #Hay datos faltantes en las variables latirtud y longitud, como bien dijimos antes

#Inspeccionamos los valores missing de "latitud" y "longitud" en el conjunto de datos de
GASOLINERAS:
View(gasolineras[which(is.na(gasolineras$latitud) & is.na(gasolineras$longitud)),])

#Imputacion de la latitud y longitud de la gasolinera de la latitud y longitud de la gasolinera
#CTRA.CHUCENA-HINOJOS (CRTA. A-481 km 0,2).
gasolineras$latitud[which(is.na(gasolineras$latitud))] = 37.349251
gasolineras$longitud[which(is.na(gasolineras$longitud))] = 6.390493

duplic11 <- which(gasolineras$direccion == "CTRA.CHUCENA-HINOJOS (CRTA. A-481 km
0,2)")
View(gasolineras[duplic11,]) #Datos imputados perfectamente

which(is.na(gasolineras$latitud))
which(is.na(gasolineras$longitud))
  #No datos missing con la imputación


#Inspeccionamos los valores missing de "latitud" y "longitud" en el conjunto de datos
#de PRECIOS_GASOL_DEF:
View(precios_gasol_def[which(is.na(precios_gasol_def$latitud) &
is.na(precios_gasol_def$longitud)),])

#Imputacion de la latitud y longitud de la gasolinera de la latitud y longitud de la gasolinera
#CTRA.CHUCENA-HINOJOS (CRTA. A-481 km 0,2).
precios_gasol_def$latitud[which(is.na(precios_gasol_def$latitud))] = 37.349251
precios_gasol_def$longitud[which(is.na(precios_gasol_def$longitud))] = 6.390493

duplic111 <- which(precios_gasol_def$direccion == "CTRA.CHUCENA-HINOJOS (CRTA.
A-481 km 0,2)")
View(precios_gasol_def[duplic111,]) #Datos imputados perfectamente
```

```
which(is.na(precios_gasol_def$latitud))
which(is.na(precios_gasol_def$longitud))
  #No datos missing con la imputación


###Tratamiento de datos atípicos
library(outliers)
grubbs.test(gasolineras$latitud, type = 10, opposite = FALSE, two.sided = TRUE) #p-value <
2.2e-16 < 0.05, 0.01, 0.1,
   #por lo que podemos afirmar que el valor 27.751944 es un outliers.
   #¿Es realmene un outliers? NO
   duplic01 <- which(gasolineras$latitud == 27.751944)
   View(gasolineras[duplic01,])

grubbs.test(gasolineras$longitud, type = 10, opposite = FALSE, two.sided = TRUE) #p-value <
2.2e-16 < 0.05, 0.01, 0.1,
   #por lo que podemos afirmar que el valor -18.011944 es un outliers.
   #¿Es realmene un outliers? NO
   duplic02 <- which(gasolineras$longitud == -18.011944)
   View(gasolineras[duplic02,])
    #En conjunto vemos que NO es un outliers ya qye se trata de Santa Cruz de Tenerife

#Filtramos por tipo de gasolina para aplicar el test:
#gasolina98
gasol98 <- which(precios_gasol_def$tipo_gasol == "gasolina98")
View(precios_gasol_def[gasol98,])
grubbs.test(precios_gasol_def[gasol98,]$media_precio, type = 10, opposite = FALSE, two.sided
= TRUE) #p-value = 3.196e-07 < 0.05, 0.01, 0.1,
   #podemos afirmar que 0.763 es un atípico
hist(precios_gasol_def[gasol98,]$media_precio)

#gasolina95Proteccion
gasol95Protec <- which(precios_gasol_def$tipo_gasol == "gasolina95Proteccion")
View(precios_gasol_def[gasol95Protec,])
grubbs.test(precios_gasol_def[gasol95Protec,]$media_precio, type = 10, opposite = FALSE,
two.sided = TRUE) #p-value = 0.002013 > 0.05, 0.01,
   #por lo que con una seguridad del 95% que el valor 0.799 no es un outliers.

#gasoleoB
gasolB <- which(precios_gasol_def$tipo_gasol == "gasoleoB")
View(precios_gasol_def[gasolB,])
grubbs.test(precios_gasol_def[gasolB,]$media_precio, type = 10, opposite = FALSE, two.sided
= TRUE) #p-value = 0.1161 > 0.05, 0.01, 0.1,
   #por lo que con una seguridad del 95% que el valor 1.1025 no es un outliers.

#gasoleoA
gasolA <- which(precios_gasol_def$tipo_gasol == "gasoleoA")
View(precios_gasol_def[gasolA,])
grubbs.test(precios_gasol_def[gasolA,]$media_precio, type = 10, opposite = FALSE, two.sided
= TRUE) #p-value = 0.01997 < 0.05, 0.01, 0.1 podemos
   #afirmar que 0.739 es un atípico.
```

```
#gasNaturalLicuado
gasNatLicuado <- which(precios_gasol_def$tipo_gasol == "gasNaturalLicuado")
View(precios_gasol_def[gasNatLicuado,])
grubbs.test(precios_gasol_def[gasNatLicuado,]$media_precio, type = 10, opposite = FALSE,
two.sided = TRUE) #p-value < 2.2e-16 < 0.05, 0.01, 0.1 podemos
   #afirmar que 0.674 es un atípico.

#gasNaturalComprimido
gasNatComprimido <- which(precios_gasol_def$tipo_gasol == "gasNaturalComprimido")
View(precios_gasol_def[gasNatComprimido,])
grubbs.test(precios_gasol_def[gasNatComprimido,]$media_precio, type = 10, opposite =
FALSE, two.sided = TRUE) #p-value < 2.2e-16 < 0.05, 0.01, 0.1 podemos
   #afirmar que 0.738 es un atípico.

#gasesLicuadosPetroleo
gasLicuadosPetrol <- which(precios_gasol_def$tipo_gasol == "gasesLicuadosPetroleo")
View(precios_gasol_def[gasLicuadosPetrol,])
grubbs.test(precios_gasol_def[gasLicuadosPetrol,]$media_precio, type = 10, opposite =
FALSE, two.sided = TRUE) #p-value = 0.9994 > 0.05, 0.01, 0.1
   #por lo que con una seguridad del 95% que el valor 0.499 no es un atípico.

#bioetanol
bioetan <- which(precios_gasol_def$tipo_gasol == "bioetanol")
View(precios_gasol_def[bioetan,])
grubbs.test(precios_gasol_def[bioetan,]$media_precio, type = 10, opposite = FALSE, two.sided
= TRUE) #p-value < 2.2e-16 < 0.05, 0.01, 0.1 podemos
   #afirmar que 1.699 es un atípico.

#biodiesel
biodios <- which(precios_gasol_def$tipo_gasol == "biodiesel")
View(precios_gasol_def[biodios,])
grubbs.test(precios_gasol_def[biodios,]$media_precio, type = 10, opposite = FALSE, two.sided
= TRUE) #p-value = 0.05218 > 0.05, 0.01,
   #por lo que con una seguridad del 95% que el valor 1.318 no es un outliers.


##¿Qué hacemos con los valores missing?: Vamos a graficar el tipo de gasolina en función del
precio, donde acortemos los valores
#gasolina98
library(ggplot2)
ggplot(precios_gasol_def[gasol98,], aes(media_precio)) +
  geom_histogram(color="white") + labs(title="Gráfico del precio medio del tipo de gasolina
gasoleo98",
                                       x ="precio medio", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#gasoleoA
ggplot(precios_gasol_def[gasolA,], aes(media_precio)) +
  geom_histogram(color="white") + labs(title="Gráfico del precio medio del tipo de gasolina
gasoleoA",
                                       x ="precio medio", y = "frecuencia") +
```

```
  theme(plot.title = element_text(hjust = 0.5))

#gasNaturalLicuado
ggplot(precios_gasol_def[gasNatLicuado,], aes(media_precio)) +
  geom_histogram(color="white") + labs(title="Gráfico del precio medio del tipo de gasolina
gasNaturalLicuado",
                                       x ="precio medio", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#gasNaturalCompimido
ggplot(precios_gasol_def[gasNatComprimido,], aes(media_precio)) +
  geom_histogram(color="white") + labs(title="Gráfico del precio medio del tipo de gasolina
gasNaturalCompimido",
                                       x ="precio medio", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#gasesLicuadosPetroleo
ggplot(precios_gasol_def[gasLicuadosPetrol,], aes(media_precio)) +
  geom_histogram(color="white") + labs(title="Gráfico del precio medio del tipo de gasolina
gasesLicuadosPetroleo",
                                       x ="precio medio", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#bioetanol
ggplot(precios_gasol_def[bioetan,], aes(media_precio)) +
  geom_histogram(color="white") + labs(title="Gráfico del precio medio del tipo de gasolina
bioetanol",
                                       x ="precio medio", y = "frecuencia") +
  theme(plot.title = element_text(hjust = 0.5))

#############################################################################
#Analisis multivariante
#############################################################################
##ANÁLISIS CLÚSTER
  #Para la media de precios:
  precio_mediaPrecio <- sample(1:nrow(precios_gasol_def), size=8697, replace=FALSE)
  precio_mediaPrecioD <- precios_gasol_def[precio_mediaPrecio, ]

  #Vamos a agrupar los tipos de gasolineras en función del precio, la latitud y longitud:
  precios_preciosCompetenc = data.frame(precios_competenciaGasoleoA$precio_5km,
precios_competenciaGasoleoA$precio_10km,
                                        precios_competenciaGasoleoA$precio_20km,
precios_competenciaGasoleoA$precio_50km,
                                        precio_mediaPrecioD$media_precio)
  colnames(precios_preciosCompetenc) <- c("precio_5km", "precio_10km", "precio_20km",
"precio_50km", "media_precio")

  #Número óptimo de cluster:
  wssplot <- function(data, nc=15, seed=1234){
    wss <- (nrow(data)-1)*sum(apply(data,2,var))
    for (i in 2:nc){
```

```
    set.seed(seed)
    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Número de clústers",
       ylab="Suma de cuadrados dentro de cada clúster")}

 wssplot(precios_preciosCompetenc, nc=6)
 wssplot(precios_preciosCompetenc, nc=15)

 #Método k-means:
 precios_preciosCompetenc_Cluster <- kmeans(precios_preciosCompetenc, 4, nstart = 20)
 precios_preciosCompetenc_Cluster

 #Gráfico de los clúster:
 library(ggplot2)
 precios_preciosCompetenc_Cluster$cluster <-
as.factor(precios_preciosCompetenc_Cluster$cluster)
 ggplot(precios_preciosCompetenc, aes(media_precio, precio_5km, color =
precios_preciosCompetenc_Cluster$cluster)) + geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.4, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1)) + geom_abline(intercept = 0, slope = 1)
 ggplot(precios_preciosCompetenc, aes(media_precio, precio_10km, color =
precios_preciosCompetenc_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.4, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1)) + geom_abline(intercept = 0, slope = 1)
 ggplot(precios_preciosCompetenc, aes(media_precio, precio_20km, color =
precios_preciosCompetenc_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.4, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1)) + geom_abline(intercept = 0, slope = 1)
 ggplot(precios_preciosCompetenc, aes(media_precio, precio_50km, color =
precios_preciosCompetenc_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.4, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1)) + geom_abline(intercept = 0, slope = 1)
 ggplot(precios_preciosCompetenc, aes(precio_5km, precio_10km, color =
precios_preciosCompetenc_Cluster$cluster))+ geom_point()
 ggplot(precios_preciosCompetenc, aes(precio_5km, precio_20km, color =
precios_preciosCompetenc_Cluster$cluster))+ geom_point()
 ggplot(precios_preciosCompetenc, aes(precio_5km, precio_50km, color =
precios_preciosCompetenc_Cluster$cluster))+ geom_point()
 ggplot(precios_preciosCompetenc, aes(precio_10km, precio_20km, color =
precios_preciosCompetenc_Cluster$cluster))+ geom_point()
 ggplot(precios_preciosCompetenc, aes(precio_10km, precio_50km, color =
precios_preciosCompetenc_Cluster$cluster))+ geom_point()
 ggplot(precios_preciosCompetenc, aes(precio_20km, precio_50km, color =
precios_preciosCompetenc_Cluster$cluster))+ geom_point()


 #Número de gasolineras en cada clúster:
 precios_preciosCompetenc_Cluster$size
 #¿En mi entorno (5km, 10km, 20km, y 50km) baja o no el precio con respecto al precio medio
de cada gasolinera?
 #5km
```

```
 cor.test(precios_competenciaGasoleoA$precio_5km, precio_mediaPrecioD$media_precio) #p-
value = 0.09341 -> no cor
 ggplot(precios_preciosCompetenc, aes(x=media_precio, y=precio_5km)) + geom_point() +
ggtitle("Gráfico de dispersión
   del precio y del de la competencia a 5 km") +
   xlab("Precio de la gasolina") + ylab("Precio de la gasolina a 5 km") +
geom_smooth(method=lm)

 #10km
 cor.test(precios_competenciaGasoleoA$precio_10km, precio_mediaPrecioD$media_precio)
#p-value = 0.1091 -> no cor
 ggplot(precios_preciosCompetenc, aes(x=media_precio, y=precio_10km)) + geom_point() +
ggtitle("Gráfico de dispersión
   del precio y del de la competencia a 10 km") +
   xlab("Precio de la gasolina") + ylab("Precio de la gasolina a 10 km") +
geom_smooth(method=lm)

 #20km
 cor.test(precios_competenciaGasoleoA$precio_20km, precio_mediaPrecioD$media_precio)
#p-value = 0.1013
 ggplot(precios_preciosCompetenc, aes(x=media_precio, y=precio_20km)) + geom_point() +
ggtitle("Gráfico de dispersión
   del precio y del de la competencia a 20 km") +
   xlab("Precio de la gasolina") + ylab("Precio de la gasolina a 20 km") +
geom_smooth(method=lm)

 #50km
 cor.test(precios_competenciaGasoleoA$precio_50km,
precio_mediaPrecioD$media_precio)#p-value = 0.09341 -> no cor
 ggplot(precios_preciosCompetenc, aes(x=media_precio, y=precio_50km)) + geom_point() +
ggtitle("Gráfico de dispersión
   del precio y del de la competencia a 50 km") +
   xlab("Precio de la gasolina") + ylab("Precio de la gasolina a 50 km") +
geom_smooth(method=lm)

 library(reshape2)
 datos = cor(precios_preciosCompetenc)
 datos.lista = melt(datos)
 names(datos.lista)=c("Variable_1","Variable_2","Correlacion")
 escala = seq(-1,1,0.1)
 (p <- ggplot(datos.lista, aes(Variable_1, Variable_2, fill=Correlacion)) +
geom_tile(aes(fill=Correlacion)) +
    scale_fill_continuous(low = "white", high = "steelblue"
                  ,breaks=escala) + theme(title="Análisis de correlaciones",
                                plot.title = element_text(face="bold", size=14)))

 #Número de gasolineras por provinicia y por clúster:
 gasolineras_clusterProvincia <- sample(1:nrow(gasolineras), size=9017, replace=FALSE)
 gasolineras_clusterProvinciaD <- gasolineras[gasolineras_clusterProvincia, ]

 cluster_media <- precios_preciosCompetenc_Cluster$cluster
```

```
 gasolineras_clusterProvinciaD$factor12 <- cluster_media
 colnames(gasolineras_clusterProvinciaD) = c("direccion", "cp", "horario", "latitud",
"longitud", "localidad",
                                    "margen", "municipio", "rotulo", "provincia", "tipo_gasol",
"número_cluster_media")

 library(sqldf)
 cluster_provincia_media = sqldf("SELECT count(*) count, provincia, número_cluster_media
                          FROM gasolineras_clusterProvinciaD group by provincia,
número_cluster_media order by número_cluster_media")

 cluster_provincia_media$número_cluster_media =
as.factor(cluster_provincia_media$número_cluster_media)
 ggplot(data=cluster_provincia_media, aes(x=provincia, y=count, fill=número_cluster_media))
+
   geom_bar(stat="identity") + coord_flip() + ggtitle("Número de gasolineras por provincias y
clúster") +
   ylab("Frecuencia") + xlab("Provincia") + guides(fill=guide_legend(title="Clúster"))



 #Para el mínimo de precios:
 precio_minPrecio <- sample(1:nrow(precios_gasol_def), size=9017, replace=FALSE)
 precio_minPrecioD <- precios_gasol_def[precio_minPrecio, ]

 #Vamos a agrupar los tipos de gasolineras en función del precio, la latitud y longitud:
 precios_preciosCompetencMin =
data.frame(min_precios_competenciaGasoleoA$precio_5km,
min_precios_competenciaGasoleoA$precio_10km,
                                    min_precios_competenciaGasoleoA$precio_20km,
min_precios_competenciaGasoleoA$precio_50km,
                                    precio_minPrecioD$media_precio)
 colnames(precios_preciosCompetencMin) <- c("precio_5km", "precio_10km", "precio_20km",
"precio_50km", "media_precio")

 #Eliminamos valores inf.:
 precios_preciosCompetencMin <-
precios_preciosCompetencMin[is.finite(rowSums(precios_preciosCompetencMin)),]

 #Número óptimo de cluster:
 wssplot <- function(data, nc=15, seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Número de clústers",
       ylab="Suma de cuadrados dentro de cada clúster")}

 wssplot(precios_preciosCompetencMin, nc=6)
 wssplot(precios_preciosCompetencMin, nc=15)
 #Método k-means:
```

```
 precios_preciosCompetencMin_Cluster <- kmeans(precios_preciosCompetencMin, 5, nstart =
20)
 precios_preciosCompetencMin_Cluster

 #Gráfico de los clúster:
 library(ggplot2)
 precios_preciosCompetencMin_Cluster$cluster <-
as.factor(precios_preciosCompetencMin_Cluster$cluster)
 ggplot(precios_preciosCompetencMin, aes(media_precio, precio_5km, color =
precios_preciosCompetencMin_Cluster$cluster)) + geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.5, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1)) + geom_abline(intercept = 0, slope = 1)
 ggplot(precios_preciosCompetencMin, aes(media_precio, precio_10km, color =
precios_preciosCompetencMin_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.5, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1)) + geom_abline(intercept = 0, slope = 1)
 ggplot(precios_preciosCompetencMin, aes(media_precio, precio_20km, color =
precios_preciosCompetencMin_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.5, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1)) + geom_abline(intercept = 0, slope = 1)
 ggplot(precios_preciosCompetencMin, aes(media_precio, precio_50km, color =
precios_preciosCompetencMin_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.5, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1)) + geom_abline(intercept = 0, slope = 1)
 ggplot(precios_preciosCompetencMin, aes(precio_5km, precio_10km, color =
precios_preciosCompetencMin_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.5, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1))
 ggplot(precios_preciosCompetencMin, aes(precio_5km, precio_20km, color =
precios_preciosCompetencMin_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.5, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1))
 ggplot(precios_preciosCompetencMin, aes(precio_5km, precio_50km, color =
precios_preciosCompetencMin_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.5, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1))
 ggplot(precios_preciosCompetencMin, aes(precio_10km, precio_20km, color =
precios_preciosCompetencMin_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.5, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1))
 ggplot(precios_preciosCompetencMin, aes(precio_10km, precio_50km, color =
precios_preciosCompetencMin_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.5, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1))
 ggplot(precios_preciosCompetencMin, aes(precio_20km, precio_50km, color =
precios_preciosCompetencMin_Cluster$cluster))+ geom_point() +
   scale_y_continuous(breaks=seq(0.7, 1.5, 0.1)) + scale_x_continuous(breaks=seq(0.4, 1.6,
0.1))


 #Número de gasolineras en cada clúster:
```

```
 precios_preciosCompetencMin_Cluster$size

 #Número de gasolineras por provinicia y por clúster:
 gasolineras_clusterProvinciaMin <- sample(1:nrow(gasolineras), size=9017, replace=FALSE)
 gasolineras_clusterProvinciaDMin <- gasolineras[gasolineras_clusterProvincia, ]

 cluster_minimo <- precios_preciosCompetencMin_Cluster$cluster
 gasolineras_clusterProvinciaDMin$factor12 <- cluster_minimo
 colnames(gasolineras_clusterProvinciaDMin) = c("direccion", "cp", "horario", "latitud",
"longitud", "localidad",
                                    "margen", "municipio", "rotulo", "provincia", "tipo_gasol",
"numero_cluster_minimo")

 library(sqldf)
 cluster_provincia_minimo = sqldf("SELECT count(*) count, provincia,
numero_cluster_minimo
                          FROM gasolineras_clusterProvinciaDMin group by provincia,
numero_cluster_minimo
                           order by numero_cluster_minimo")

 cluster_provincia_minimo$numero_cluster_minimo =
as.factor(cluster_provincia_minimo$numero_cluster_minimo)
 ggplot(data=cluster_provincia_minimo, aes(x=provincia, y=count,
fill=numero_cluster_minimo)) +
   geom_bar(stat="identity") + coord_flip() + ggtitle("Número de gasolineras por provincias y
clúster") +
   ylab("Frecuencia") + xlab("Provincia") + guides(fill=guide_legend(title="Clúster")) +
   scale_fill_manual(values=c("red", "green", "gree n4", "blue", "purple"))

#########################################################################
#Modelos predictivos
#########################################################################
#Nuestro objetivo es predecir el precio de las gasolinas en función de una serie de variables,
#como son, la dirección, el cp, entre otras, de la gasolinera.

#División del conjunto de datos:
library(caTools)
sample = sample.split(precios_gasol_def, SplitRatio = 0.7)
precios_gasol_def_entren = subset(precios_gasol_def, sample == TRUE) #modelos
write.table(precios_gasol_def_entren,file="C:/Users/Beatriz/Desktop/TFM_def/precios_gasol_d
ef_entren.csv", sep=";", quote = FALSE,
          row.names=FALSE, dec = ",")

sample1 = sample.split(precios_gasol_def, SplitRatio = 0.3)
precios_gasol_def_prueba = subset(precios_gasol_def, sample1 == TRUE) #predicción
write.table(precios_gasol_def_prueba,file="C:/Users/Beatriz/Desktop/TFM_def/precios_gasol_
def_prueba.csv", sep=";", quote = FALSE,
          row.names=FALSE, dec = ",")

library(h2o)
h2o.init(nthreads=-1, max_mem_size="10G")
```

```
entren = h2o.importFile(path =
normalizePath("C:/Users/Beatriz/Desktop/TFM_def/precios_gasol_def_entren.csv"))
prueba = h2o.importFile(path =
normalizePath("C:/Users/Beatriz/Desktop/TFM_def/precios_gasol_def_prueba.csv"))

##Preparación del conjunto de datos:
#a) EXtraemos variables continuas y categóricas
continuas = c("latitud", "longitud", "media_precio")
categor = c("horario","rotulo","provincia", "tipo_gasol")
#Extraemos variables continuas sin var.objetivo
continuasin <- c("latitud", "longitud")
precios <- precios_gasol_def_entren[,c(continuas,categor)]
#b) Creamos dummys a las variables categóricas
precios$horario = sapply(as.character(precios$horario), switch, "no abre las 24 horas" = 1,
"abierto las 24 horas" = 2,
                         USE.NAMES = F)
precios$horario = as.factor(precios$horario)
levels(precios$horario) <- relevel(precios$horario, ref = "no abre las 24 horas")
levels(precios$rotulo) <- relevel(precios$rotulo, ref = "PETRONOR")
levels(precios$provincia) <- relevel(precios$provincia, ref = "CIUDAD REAL")
levels(precios$tipo_gasol) <- relevel(precios$tipo_gasol, ref = "gasolina95Proteccion")

precios_dumm <- data.frame(precios$latitud, precios$longitud, precios$media_precio,
precios$horario,
                           precios$rotulo,
                           precios$provincia, precios$tipo_gasol)
colnames(precios_dumm) <- c("latitud","longitud",
"media_precio","horario","rotulo","provincia",
                            "tipo_gasol")
#c)Estandarizamos las variables continuas, excepto la dependiente
#Calculamos medias y dtipica de datos y estandarizamos (solo las continuas)
means <- apply(precios_dumm[,continuas],2,mean)
sds <- sapply(precios_dumm[,continuas],sd)
#Estandarizamos las continuas y uno con las categóricas
precios2 <- scale(precios_dumm[,continuas], center = means, scale = sds)
numerocont <- which(colnames(precios_dumm)%in%continuas)
precios2 <-cbind(precios2, precios_dumm[,-numerocont])


##########################REGRESIÓN LINEAL
# ****************************************
# Funciones previas
# ****************************************
kfcv.sizes = function(n, k=10) {
  sizes = c()
  for (i in 1:k) {
    first = 1 + (((i - 1) * n) %/% k)
    last = ((i * n) %/% k)
    sizes = append(sizes, last - first + 1)
  }
  sizes
```

```
}

kfcv.testing = function(n, k=4) {
  indices = list()
  sizes = kfcv.sizes(n, k=k)
  values = 1:n
  for (i in 1:k) {
    # take a random sample of given size
    s = sample(values, sizes[i])
    # append random sample to list of indices
    indices[[i]] = s
    # remove sample from values
    values = setdiff(values, s)
  }
  indices
}

# ****************************************
# Función general CV LINEAL
# Lo más cómodo es generar un data frame con solo las
# variables input que se van a utilizar, y la output
# Para comparar con redes se hace todo estandarizando y después deshaciendo
# ****************************************

valcruzalin= function(data, k,vardep,semilla) {
  set.seed(semilla)
  result = list()
  alltestingindices = kfcv.testing(dim(data)[1],k)
  for (i in 1:k) {
    testingindices = alltestingindices[[i]]
    train = data[-testingindices,]
    test= data[testingindices,]

    # Primero aplicar el algoritmo a datos train, es necesario aplicar
    # la sintaxix correcta para cada algoritmo/paquete.

    # Ejemplo con regresión lineal
    formula1<-as.formula(paste(vardep,"~."))
    algobjeto<-lm(data=train,formula1)
    result[[i]]<-predict.lm(algobjeto,test)

  }
  result<-data.frame(unlist(result))
  nombres<-unlist(alltestingindices)
  rownames(result)<-nombres
  result<-merge(data[,vardep],result, by="row.names", all=TRUE)
  result$error<-(result$x-result$unlist.result.)^2

  mediaerror<-mean(result$error)
  return(list(result,mediaerror))
}
```

```
# Bucle para validación cruzada repetida
# Para conservar el valor promedio de error por cada ejecución de Validación cruzada

cvrepetidalin<-function(data, k,vardep,inicio,numero){
  error<-numeric()
  for (i in 1:numero){
    indice<-i+inicio-1
    error[i]<-valcruzalin(data, k,vardep,indice)[[2]]
  }
  error<-as.data.frame(error)
  return(error)
}


#Selección de variables
library(MASS)
modelo <- lm(media_precio~.-1, data = precios2)
#StepWise
StepWise <- stepAIC(modelo, direction="both")
s <- StepWise[[1]]
dput(names(s))
variables.StepWise = c(dput(names(s)))
variables.seleccion = variables.StepWise
#Reconstruimos el archivo con solo las variables de interés
media_precio <- precios2[,"media_precio"]
precios2StepWise <- cbind(precios2[,variables.StepWise],media_precio)
#Forward
Forward <- stepAIC(modelo, direction="forward")
f <- Forward[[1]]
dput(names(f))
variables.Forward = c(dput(names(f)))
variables.seleccion = variables.Forward
#Reconstruimos el archivo con solo las variables de interés
media_precio <- precios2[,"media_precio"]
precios2Forward <- cbind(precios2[,variables.Forward],media_precio)
#Backward
Backward <- stepAIC(modelo, direction="backward")
f <- Backward[[1]]
dput(names(f))
variables.Backward = c(dput(names(f)))
variables.seleccion = variables.Backward
#Reconstruimos el archivo con solo las variables de interés
media_precio <- precios2[,"media_precio"]
precios2Backward <- cbind(precios2[,variables.Backward],media_precio)
lin <- cvrepetidalin(precios2,4,"media_precio",12346,5) #data, k,vardep,inicio,numero
lin1 <- cvrepetidalin(precios2,5,"media_precio",12346,5)
lin2 <- cvrepetidalin(precios2,3,"media_precio",12346,5)
lin3 <- cvrepetidalin(precios2,2,"media_precio",12346,5)
lin4 <- cvrepetidalin(precios2,6,"media_precio",12346,5)
lin5 <- cvrepetidalin(precios2,4,"media_precio",12349,5) #distinta semilla
```

```
lin6 <- cvrepetidalin(precios2,5,"media_precio",12349,5)
lin7 <- cvrepetidalin(precios2,3,"media_precio",12349,5)
lin8 <- cvrepetidalin(precios2,2,"media_precio",12349,5)
lin9 <- cvrepetidalin(precios2,6,"media_precio",12349,5)

#Comparación de modelos en boxplot:
#Union de los errores
uni <- as.data.frame(cbind(lin=lin$error*0.1793881,lin1=lin1$error*0.1793881,
lin2=lin2$error*0.1793881, lin3=lin3$error*0.1793881,
                    lin4=lin4$error*0.1793881,lin5=lin5$error*0.1793881,
lin6=lin6$error*0.1793881, lin7=lin7$error*0.1793881,
                    lin8=lin8$error*0.1793881,lin9=lin9$error*0.1793881))

# Recolocamos el archivo
library(reshape)
parabox<-melt(uni)
boxplot(data=parabox,value~variable, main = "Modelos de regresión lineal", xlab = "modelo",
ylab = "error")


##########################RED NEURONAL
# ****************************************
# Funci???n general CV RED NNET
# Lo m???s c???modo es generar un data frame con solo las
# variables input que se van a utilizar, y la output
# Para comparar con redes se hace todo estandarizando y despu???s deshaciendo
# ****************************************
library(nnet)
library(h2o)

valcruzannet= function(data, k,vardep,semilla,nNodos,mIteracciones) {
  set.seed(semilla)
  result = list()
  alltestingindices = kfcv.testing(dim(data)[1],k)
  for (i in 1:k) {
    testingindices = alltestingindices[[i]]
    train1 = data[-testingindices,]
    test1= data[testingindices,]
    means <- apply(train1, 2, mean)
    sds<- sapply(train1,sd)
    train <- as.data.frame(scale(train1, center = means, scale = sds))
    test<- as.data.frame(scale(test1, center = means, scale = sds))

    # Primero aplicar el algoritmo a datos train, es necesario aplicar
    # la sintaxix correcta para cada algoritmo/paquete.
    # Ejemplo con nnet: necesario previamente estandarizar y deshacer después
    formula1<-as.formula(paste(vardep,"~."))
    #algobjeto<-nnet(data=train,formula1,linout = TRUE,size=30,maxit=100)
    algobjeto<-nnet(data=train,formula1,linout = TRUE,size=nNodos,maxit=mIteracciones)
    result[[i]]<-predict(algobjeto,newdata=test,type="raw")
    result[[i]]<-result[[i]]*sd(train1[,vardep])+mean(train1[,vardep])
```

```
  }

  result<-data.frame(unlist(result))
  nombres<-unlist(alltestingindices)
  rownames(result)<-nombres
  result<-merge(data[,vardep],result, by="row.names", all=TRUE)
  result$error<-(result$x-result$unlist.result.)^2

  mediaerror<-mean(result$error)
  return(list(result,mediaerror))
}

# Bucle para validaci???n cruzada repetida
# Para conservar el valor promedio de error por cada ejecuci???n de Validaci???n cruzada

cvrepetidannet<-function(data, k,vardep,inicio,numero,nNodos,mIteracciones){ #K numero de
grupos valid. cruzada, numero: numero de semillas diferentes
  error<-numeric()
  for (i in 1:numero){
    indice<-i+inicio-1
    error[i]<-valcruzannet(data, k,vardep,indice,nNodos,mIteracciones)[[2]]
  }
  error<-as.data.frame(error)
  return(error)
}

# ****************************************
# Funciónn general CV RED H2O
# Lo más cómodo es generar un data frame con solo las
# variables input que se van a utilizar, y la output
# Para comparar con redes se hace todo estandarizando y después deshaciendo
# ****************************************

valcruzah2o= function(data, k,vardep,semilla,nNodos,mIteracciones,activacion,optimizacion) {

  # Es necesario reordenar las columnas en data frame antes

  data<- data[, c(vardep, setdiff(names(data), vardep))]
  numerocol<-ncol(data)

  set.seed(semilla)
  result = list()
  alltestingindices = kfcv.testing(dim(data)[1],k)
  for (i in 1:k) {
    testingindices = alltestingindices[[i]]
    train1 = data[-testingindices,]
    test1= data[testingindices,]
    means <- apply(train1, 2, mean)
    sds<- sapply(train1,sd)
    train <- as.data.frame(scale(train1, center = means, scale = sds))
```

```
    test<- as.data.frame(scale(test1, center = means, scale = sds))

    # Primero aplicar el algoritmo a datos train, es necesario aplicar
    # la sintaxix correcta para cada algoritmo/paquete.

    train.hex <- as.h2o(train, destination_frame = "train.hex")
    test.hex <- as.h2o(test, destination_frame = "test.hex")

    red5<-h2o.deeplearning(x = 2:numerocol,y=1,training_frame = train.hex,
                           #hidden = c(30),epochs =100,adaptive_rate=FALSE,rate=0.001,
                           hidden = c(nNodos),epochs =mIteracciones,adaptive_rate=FALSE,rate=0.001,
                           train_samples_per_iteration=0,activation = activacion,
nesterov_accelerated_gradient = optimizacion)

    predi5<- h2o.predict(red5,test.hex)
    predi5= as.data.frame(predi5)
    pred<-as.vector(predi5$predict)
    result[[i]]<-pred*sd(train1[,vardep])+mean(train1[,vardep])
   }
  result<-data.frame(unlist(result))
  nombres<-unlist(alltestingindices)
  rownames(result)<-nombres
  result<-merge(data[,vardep],result, by="row.names", all=TRUE)
  result$error<-(result$x-result$unlist.result.)^2

  mediaerror<-mean(result$error)
  return(list(result,mediaerror))
}

# Bucle para validaci???n cruzada repetida
# Para conservar el valor promedio de error por cada ejecuci???n de Validaci???n cruzada

cvrepetidah2o<-function(data, k,vardep,inicio,numero,nNodos,mIteracciones, activacion,
optimizacion){
  h2o.init()
  error<-numeric()
  for (i in 1:numero){
    indice<-i+inicio-1
    error[i]<-valcruzah2o(data,
k,vardep,indice,nNodos,mIteracciones,activacion,optimizacion)[[2]]
   }
  error<-as.data.frame(error)
  return(error)
}

red1 <- cvrepetidannet(precios2,4,"media_precio",12346, 5, 5, 10) #data,
k,vardep,inicio,numero,nNodos,mIteracciones
red2 <- cvrepetidah2o(precios2,4,"media_precio",12346,5, 8, 20, "Tanh") #data,
k,vardep,inicio,numero,nNodos,mIteracciones, activacion, optimizacion
red3 <- cvrepetidah2o(precios2,5,"media_precio",12346,5, 10, 30, "Tanh")
red4 <- cvrepetidah2o(precios2,3,"media_precio",12346,5, 12, 40, "TanhWithDropout")
```

```
red5 <- cvrepetidah2o(precios2,3,"media_precio",12346,5, 13, 50, "TanhWithDropout")
red6 <- cvrepetidah2o(precios2,6,"media_precio",12346,5, 15, 60, "Maxout")
red7 <- cvrepetidah2o(precios2,4,"media_precio",12346,5, 17, 70, "Maxout")
red8 <- cvrepetidannet(precios2,5,"media_precio",12349, 5, 5, 10) #distinta semilla
red9 <-  cvrepetidah2o(precios2,3,"media_precio",12349,5, 8, 20, "Tanh")
red10 <- cvrepetidah2o(precios2,3,"media_precio",12349,5, 10, 30, "Tanh")
red11 <- cvrepetidah2o(precios2,2,"media_precio",12349,5, 12, 40, "TanhWithDropout")
red12 <- cvrepetidah2o(precios2,6,"media_precio",12349,5, 13, 50, "TanhWithDropout")
red13 <- cvrepetidah2o(precios2,6,"media_precio",12349,5, 15, 60, "Maxout")
red14 <- cvrepetidah2o(precios2,6,"media_precio",12349,5, 17, 70, "Maxout")

#Comparación por boxplot
#Union de los errores
uni <-
as.data.frame(cbind(red1=red1$error*0.1793881,h2o2=red2$error*0.1793881,h2o3=red3$error
*0.1793881, h2o4=red4$error*0.1793881,

red5=red5$error*0.1793881,h2o6=red6$error*0.1793881,h2o7=red7$error*0.1793881,
h2o8=red8$error*0.1793881,

red9=red9$error*0.1793881,h2o10=red10$error*0.1793881,h2o11=red11$error*0.1793881,
h2o12=red12$error*0.1793881,
                    h2o13=red13$error*0.1793881, h2o14=red14$error*0.1793881))

#Vemos que hay modelos que no se ven claramente por lo que separamos los modelos que se
parecen
uni1 <- as.data.frame(cbind(h2o4=red4$error*0.1793881, red5=red5$error*0.1793881,
h2o11=red11$error*0.1793881, h2o12=red12$error*0.1793881))
uni2 <-
as.data.frame(cbind(red1=red1$error*0.1793881,h2o2=red2$error*0.1793881,h2o3=red3$error
*0.1793881,
                     h2o6=red6$error*0.1793881,h2o7=red7$error*0.1793881,
h2o8=red8$error*0.1793881,
                     red9=red9$error*0.1793881,h2o10=red10$error*0.1793881,
                     h2o13=red13$error*0.1793881, h2o14=red14$error*0.1793881))

#Recolocamos el archivo
library(reshape)
parabox <- melt(uni)
boxplot(data=parabox, value~variable, main = "Modelos de red neuronal", xlab = "modelo",
ylab = "error")

parabox <- melt(uni1)
boxplot(data=parabox, value~variable, main = "Modelos 4, 5, 11 y 12 de red neuronal", xlab =
"modelo", ylab = "error")
parabox <- melt(uni2)
boxplot(data=parabox, value~variable, main = "Demás modelos de red neuronal", xlab =
"modelo", ylab = "error")


##########################RANDOM FOREST
```

```
# ****************************************
# Funciones previas
# ****************************************
library(randomForest)
library(caret)
library(e1071)

kfcv.sizes = function(n, k=10) {
  sizes = c()
  for (i in 1:k) {
    first = 1 + (((i - 1) * n) %/% k)
    last = ((i * n) %/% k)
    sizes = append(sizes, last - first + 1)
  }
  sizes
}

kfcv.testing = function(n, k=4) {
  indices = list()
  sizes = kfcv.sizes(n, k=k)
  values = 1:n
  for (i in 1:k) {
    # take a random sample of given size
    s = sample(values, sizes[i])
    # append random sample to list of indices
    indices[[i]] = s
    # remove sample from values
    values = setdiff(values, s)
  }
  indices
}

#************************************************************
# Ejemplo con rf random forest
# n.trees es el nÃºmero de iteraciones, mtry el nÃºmero de variables
# a sortear en cada nodo, sampsize el tamaÃ±o de muestra,maxnodes el numero de hojas
#************************************************************

valcruzarflog= function(data, k,vardep,semilla, niteracciones, nobserXnodo, nvariables) {
  set.seed(semilla)
  result = list()
  alltestingindices = kfcv.testing(dim(data)[1],k)
  for (i in 1:k) {
    testingindices = alltestingindices[[i]]
    train = data[-testingindices,]
    test= data[testingindices,]

    formula1<-as.formula(paste("factor(",vardep,")","~."))

    #rfGrid <- expand.grid(mtry=c(6))
    rfGrid <- expand.grid(mtry=c(nvariables))
```

```
   control <- trainControl(method = "none")

   rf.caret <- train(formula1, data=train,trControl=control,
                  method="rf",maxnodes=20,
                  #n.trees = c(200),nodesize= 10,
                  n.trees = c(niteracciones),nodesize= nobserXnodo,
                  sampsize=200,
                  distribution="bernoulli",tuneGrid=rfGrid)
   result[[i]]<-predict(object=rf.caret,newdata=test)
  }
  result<-data.frame(unlist(result))
  nombres<-unlist(alltestingindices)
  rownames(result)<-nombres
  result<-merge(data[,vardep],result, by="row.names", all=TRUE)

  confusion<-table(result$x,result$unlist.result.)

  error<-1-sum(diag(confusion))/sum(confusion)

  return(list(result,error))

}

#*****************************************************
# Ejemplo con gbm random forest
# n.trees es el número de iteraciones, mtry el número de variables
# a sortear en cada nodo, sampsize el tamaño de muestra,maxnodes el numero de hojas
#*****************************************************

valcruzarf= function(data, k,vardep,semilla,niteracciones,nobserXnodo,nvariables) {
  set.seed(semilla)
  result = list()
  alltestingindices = kfcv.testing(dim(data)[1],k)
  for (i in 1:k) {
    testingindices = alltestingindices[[i]]
    train = data[-testingindices,]
    test= data[testingindices,]

    formula1<-as.formula(paste("factor(",vardep,")","~."))

    #rfGrid <- expand.grid(mtry=c(6))
    rfGrid <- expand.grid(mtry=c(nvariables))
    control <- trainControl(method = "none")

    rf.caret <- train(formula1, data=train,trControl=control,
                  method="rf",maxnodes=20,
                  #n.trees = c(200),nodesize= 10,
                  n.trees = c(niteracciones),nodesize= nobserXnodo,
                  sampsize=200,
                  distribution="bernoulli",tuneGrid=rfGrid)
    result[[i]]<-predict(object=rf.caret,newdata=test)
```

```
  }
  result<-data.frame(unlist(result))
  nombres<-unlist(alltestingindices)
  rownames(result)<-nombres
  result<-merge(data[,vardep],result, by="row.names", all=TRUE)

  confusion<-table(result$x,result$unlist.result.)

  error<-1-sum(diag(confusion))/sum(confusion)

  return(list(result,error))

}

#*****************************************************************
# Bucle para validación cruzada repetida
# Para conservar el valor promedio de error por cada ejecución de Validación cruzada
#*****************************************************************

cvrepetidarf<-function(data,k,vardep,inicio,numero,niteracciones,nobserXnodo,nvariables){
  error<-numeric()
  for (i in 1:numero){
    indice<-i+inicio-1
    error[i]<-valcruzarflog(data, k,vardep,indice,niteracciones , nobserXnodo, nvariables)[[2]]
  }
  error<-as.data.frame(error)
  return(error)
}

h2o.init(nthreads=-1, max_mem_size="13G")

#En este caso con una muestra más pequeña ya que sino por motivos de memoria en la
computación
precios22 <- sample(1:nrow(precios2), size=10000, replace=FALSE)
precios2muestra <- precios2[precios22, ]

rf <- cvrepetidarf(precios2muestra,4,"media_precio",12346,5, 10, 5, 7)
#data,k,vardep,inicio,numero,niteracciones,nobserXnodo,nvariables
rf1 <- cvrepetidarf(precios2muestra,3,"media_precio",12346,5, 20, 5, 7)
rf2 <- cvrepetidarf(precios2muestra,5,"media_precio",12346,5, 30, 5, 7)
rf3 <- cvrepetidarf(precios2muestra,6,"media_precio",12346,5, 40, 6, 7)
rf4 <- cvrepetidarf(precios2muestra,3,"media_precio",12346,5, 50, 2, 7)
rf5 <- cvrepetidarf(precios2muestra,4,"media_precio",12346,5, 60, 3, 7)
rf6 <- cvrepetidarf(precios2muestra,4,"media_precio",12346,5, 70, 3, 7)
rf7 <- cvrepetidarf(precios2muestra,4,"media_precio",12349,5, 10, 5, 7) #distinta semilla
rf8 <- cvrepetidarf(precios2muestra,3,"media_precio",12349,5, 20, 5, 7)
rf9 <- cvrepetidarf(precios2muestra,5,"media_precio",12349,5, 30, 5, 7)
rf10 <- cvrepetidarf(precios2muestra,6,"media_precio",12349,5, 40, 6, 7)
rf11 <- cvrepetidarf(precios2muestra,3,"media_precio",12349,5, 50, 2, 7)
rf12 <- cvrepetidarf(precios2muestra,4,"media_precio",12349,5, 60, 3, 7)
rf13 <- cvrepetidarf(precios2muestra,4,"media_precio",12349,5, 70, 3, 7)
```

```
#Comparación por boxplot
#Union de los errores
uni <-
as.data.frame(cbind(rf=rf$error*0.1793881,rf1=rf1$error*0.1793881,rf2=rf2$error*0.1793881,
rf3=rf3$error*0.1793881,
                    rf4=rf4$error*0.1793881,rf5=rf5$error*0.1793881,rf6=rf6$error*0.1793881,
rf7=rf7$error*0.1793881,

rf8=rf8$error*0.1793881,rf9=rf9$error*0.1793881,rf10=rf10$error*0.1793881,
rf11=rf11$error*0.1793881,
                    rf12=rf12$error*0.1793881,rf13=rf13$error*0.1793881))

#Vemos que hay modelos que no se ven claramente por lo que separamos los modelos que se
parecen
uni1 <- as.data.frame(cbind(rf4=rf4$error*0.1793881, rf9=rf9$error*0.1793881))
uni2 <-
as.data.frame(cbind(rf=rf$error*0.1793881,rf1=rf1$error*0.1793881,rf2=rf2$error*0.1793881,
rf3=rf3$error*0.1793881,
                     rf5=rf5$error*0.1793881,rf6=rf6$error*0.1793881, rf7=rf7$error*0.1793881,
                     rf8=rf8$error*0.1793881,rf10=rf10$error*0.1793881,
rf11=rf11$error*0.1793881,
                     rf12=rf12$error*0.1793881,rf13=rf13$error*0.1793881))

#Recolocamos el archivo
library(reshape)
parabox <- melt(uni)
boxplot(data=parabox, value~variable, main = "Modelos de random forest", xlab = "modelo",
ylab = "error")

parabox <- melt(uni1)
boxplot(data=parabox, value~variable, main = "Modelos 4 y 9 de random forest", xlab =
"modelo", ylab = "error")

parabox <- melt(uni2)
boxplot(data=parabox, value~variable, main = "Modelos 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, y 13
      de random forest", xlab = "modelo", ylab = "error")


##########################SMV
#
#
#
library(e1071)

#En este caso con una muestra más pequeña ya que sino por motivos de tiempos en la ejecución
(la misma que antes)
precios22 <- sample(1:nrow(precios2), size=10000, replace=FALSE)
precios2muestra <- precios2[precios22, ]

x <- cbind(precios2muestra$latitud, precios2muestra$longitud, precios2muestra$horario,
```

```
        precios2muestra$rotulo, precios2muestra$provincia,
         precios2muestra$tipo_gasol)
y <- precios2muestra$media_precio

#Kernel lineal
set.seed(12346)
svmlineal <- svm(x, y, type="eps-regression", kernel="linear",
              epsilon=0.01, cost=10, scale=F, cross = 3)
svmlineal1 <- svm(x, y, type="eps-regression", kernel="linear",
               epsilon=0.02, cost=15, scale=F, cross = 4)
svmlineal2 <- svm(x, y, type="eps-regression", kernel="linear",
               epsilon=0.01, cost=20, scale=F, cross = 5)
svmlineal3 <- svm(x, y, type="eps-regression", kernel="linear",
               epsilon=0.02, cost=25, scale=F, cross = 6)
svmlineal4 <- svm(x, y, type="eps-regression", kernel="linear",
               epsilon=0.01, cost=30, scale=F, cross = 3)
set.seed(12349)
svmlineal5 <- svm(x, y, type="eps-regression", kernel="linear",
               epsilon=0.01, cost=10, scale=F, cross = 3) #distinta semilla
svmlineal6 <- svm(x, y, type="eps-regression", kernel="linear",
               epsilon=0.02, cost=15, scale=F, cross = 3)
svmlineal7 <- svm(x, y, type="eps-regression", kernel="linear",
               epsilon=0.01, cost=20, scale=F, cross = 3)
svmlineal8 <- svm(x, y, type="eps-regression", kernel="linear",
               epsilon=0.02, cost=25, scale=F, cross = 3)
svmlineal9 <- svm(x, y, type="eps-regression", kernel="linear",
               epsilon=0.01, cost=30, scale=F, cross = 3)

#Kernel no lineal
set.seed(12346)
svmradial6 <- svm(x,y, type="eps-regression", kernel="radial",
               epsilon=0.01, gamma=40, cost=5, scale=F, cross = 3)
svmradial7 <- svm(x,y, type="eps-regression", kernel="radial",
               epsilon=0.02, gamma=40, cost=10, scale=F, cross = 3)
svmradial8 <- svm(x,y, type="eps-regression", kernel="radial",
               epsilon=0.01, gamma=40, cost=15, scale=F, cross = 3)
svmradial9 <- svm(x,y, type="eps-regression", kernel="radial",
               epsilon=0.02, gamma=40, cost=20, scale=F, cross = 3)
svmradial10 <- svm(x,y, type="eps-regression", kernel="radial",
                epsilon=0.01, gamma=40, cost=30, scale=F, cross = 2)
set.seed(12349)
svmradial11 <- svm(x,y, type="eps-regression", kernel="radial",
                epsilon=0.01, gamma=40, cost=5, scale=F, cross = 3) #distinta semilla
svmradial12 <- svm(x,y, type="eps-regression", kernel="radial",
                epsilon=0.02, gamma=40, cost=10, scale=F, cross = 4)
svmradial13 <- svm(x,y, type="eps-regression", kernel="radial",
                epsilon=0.01, gamma=40, cost=15, scale=F, cross = 5)
svmradial14 <- svm(x,y, type="eps-regression", kernel="radial",
                epsilon=0.02, gamma=40, cost=20, scale=F, cross = 6)
svmradial15 <- svm(x,y, type="eps-regression", kernel="radial",
                epsilon=0.01, gamma=40, cost=30, scale=F, cross = 2)
```

```
#Comparación por boxplot
#Union de los errores: semilla 12346
uni <-
as.data.frame(cbind(svm=abs(svmlineal$residuals*0.1793881),svm1=abs(svmlineal1$residuals
*0.1793881),svm2=abs(svmlineal2$residuals*0.1793881),
                    svm3=abs(svmlineal3$residuals*0.1793881),

svm4=abs(svmlineal4$residuals*0.1793881),svmr6=abs(svmradial6$residuals*0.1793881),
svmr7=abs(svmradial7$residuals*0.1793881),

svmr8=abs(svmradial8$residuals*0.1793881),svmr9=abs(svmradial9$residuals*0.1793881),sv
mr10=abs(svmradial10$residuals*0.1793881)))

#semilla 12349
uni1 <-
as.data.frame(cbind(svm5=abs(svmlineal5$residuals*0.1793881),svm16=abs(svmlineal6$residu
als*0.1793881),svm7=abs(svmlineal7$residuals*0.1793881),
  svm8=abs(svmlineal8$residuals*0.1793881),svm9=abs(svmlineal9$residuals*0.1793881),

svmr11=abs(svmradial11$residuals*0.1793881),svmr12=abs(svmradial12$residuals*0.1793881
),svmr13=abs(svmradial13$residuals*0.1793881),

svmr114=abs(svmradial14$residuals*0.1793881),svmr15=abs(svmradial15$residuals*0.179388
1)))

#Recolocamos el archivo
library(reshape)
parabox <- melt(uni)
boxplot(data=parabox, value~variable, main = "Modelos de svm lineal y radial semilla 12346",
xlab = "modelo", ylab = "error", ylim = c(0:1))

parabox <- melt(uni1)
boxplot(data=parabox, value~variable, main = "Modelos de svm lineal y radial semilla 12349",
xlab = "modelo", ylab = "error", ylim = c(0:1))

#Como no se ven bien, los separamos
uni2 <-
as.data.frame(cbind(svm=abs(svmlineal$residuals*0.1793881),svm1=abs(svmlineal1$residuals
*0.1793881),svm2=abs(svmlineal2$residuals*0.1793881),
                    svm3=abs(svmlineal3$residuals*0.1793881),
svm4=abs(svmlineal4$residuals*0.1793881)))

uni3 <- as.data.frame(cbind(svmr6=abs(svmradial6$residuals*0.1793881),
svmr7=abs(svmradial7$residuals*0.1793881),
svmr8=abs(svmradial8$residuals*0.1793881),svmr9=abs(svmradial9$residuals*0.1793881),sv
mr10=abs(svmradial10$residuals*0.1793881)))

uni4 <-
as.data.frame(cbind(svm5=abs(svmlineal5$residuals*0.1793881),svm16=abs(svmlineal6$residu
als*0.1793881),svm7=abs(svmlineal7$residuals*0.1793881),
```

```
svm8=abs(svmlineal8$residuals*0.1793881),svm9=abs(svmlineal9$residuals*0.1793881)))

uni5 <-
as.data.frame(cbind(svmr11=abs(svmradial11$residuals*0.1793881),svmr12=abs(svmradial12$
residuals*0.1793881),svmr13=abs(svmradial13$residuals*0.1793881),

svmr114=abs(svmradial14$residuals*0.1793881),svmr15=abs(svmradial15$residuals*0.179388
1)))

#Recolocamos el archivo
parabox <- melt(uni2)
boxplot(data=parabox, value~variable, main = "Modelos de svm lineal semilla 12346", xlab =
"modelo", ylab = "error", ylim = c(0:1))

parabox <- melt(uni3)
boxplot(data=parabox, value~variable, main = "Modelos de svm radial semilla 12346", xlab =
"modelo", ylab = "error", ylim = c(0:1))

parabox <- melt(uni4)
boxplot(data=parabox, value~variable, main = "Modelos de svm lineal semilla 12349", xlab =
"modelo", ylab = "error", ylim = c(0:1))

parabox <- melt(uni5)
boxplot(data=parabox, value~variable, main = "Modelos de svm radial semilla 12349", xlab =
"modelo", ylab = "error", ylim = c(0:1))

#########################################################################
#Comparación de modelos
#########################################################################
uni <- as.data.frame(cbind(lin1=lin1$error*0.1793881, h2o7=red7$error*0.1793881,
rf1=rf1$error*0.1793881, svm=abs(svmlineal$residuals*0.1793881)))

# Recolocamos el archivo
library(reshape)
parabox<-melt(uni)
boxplot(data=parabox,value~variable, main = "Mejores modelos de predicción", xlab =
"modelo", ylab = "error", ylim = c(0:1))

#Como no se ven bien, los separamos
uni1 <- as.data.frame(cbind(lin1=lin1$error*0.1793881, h2o7=red7$error*0.1793881,
rf1=rf1$error*0.1793881))
uni2 <- as.data.frame(cbind(svm=abs(svmlineal$residuals*0.1793881)))

parabox <- melt(uni1)
boxplot(data=parabox, value~variable, main = "Mejor modelo de reg, red, y rf", xlab =
"modelo", ylab = "error", ylim = c(0:1))

parabox <- melt(uni2)
boxplot(data=parabox, value~variable, main = "Mejor modelo de svm lineal", xlab = "modelo",
ylab = "error", ylim = c(0:1))
```

```
#########mejor modelo el de svm

#Interpretación del modelo
###Sacamos las predicciones del modelo de svm
library(caTools)
sample1 = sample.split(precios_gasol_def, SplitRatio = 0.3)
precios_gasol_def_prueba = subset(precios_gasol_def, sample1 == TRUE) #predicción
write.table(precios_gasol_def_prueba,file="C:/Users/Beatriz/Desktop/TFM_def/precios_gasol_
def_prueba.csv", sep=";", quote = FALSE,
          row.names=FALSE, dec = ",")

#Creamos un conjunto de datos sin las variables que no foramn parte del modelo y sin la
variable predictor:
precios_gasol_def_prueba_def <- data.frame(precios_gasol_def_prueba[, c(-1, -2, -7, -9, -12)])
precios_gasol_def_prueba_def <- data.frame(precios_gasol_def_prueba_def[, c(-1, -5)])

#Preparación del conjunto de datos (tiene que estar en la misma escala que el conjunto de datos
del modelo):
#a) EXtraemos variables continuas y categóricas
continuas = c("latitud", "longitud")
categor = c("horario", "rotulo","provincia", "tipo_gasol")
#Extraemos variables continuas sin var.objetivo
continuasin <- c("latitud", "longitud")
prueba <- precios_gasol_def_prueba_def[,c(continuas,categor)]
#b) Creamos dummys a las variables categóricas
prueba$horario = sapply(as.character(prueba$horario), switch, "no abre las 24 horas" = 1,
"abierto las 24 horas" = 2,
                    USE.NAMES = F)
levels(prueba$rotulo) <- relevel(prueba$rotulo, ref = "otros")
levels(prueba$provincia) <- relevel(prueba$provincia, ref = "BARCELONA")
levels(prueba$tipo_gasol) <- relevel(prueba$tipo_gasol, ref = "gasolina95Proteccion")
prueba_dumm <- data.frame(prueba$latitud, prueba$longitud, as.numeric(prueba$horario),
                      as.numeric(prueba$rotulo),
                      as.numeric(prueba$provincia), as.numeric(prueba$tipo_gasol))
colnames(prueba_dumm) <- c("latitud","longitud","horario","rotulo","provincia",
                       "tipo_gasol")
#c)Estandarizamos las variables continuas, excepto la dependiente
#Calculamos medias y dtipica de datos y estandarizamos (solo las continuas)
means <- apply(prueba_dumm[,continuas],2,mean)
sds <- sapply(prueba_dumm[,continuas],sd)
#Estandarizamos las continuas y uno con las categóricas
prueba2 <- scale(prueba_dumm[,continuas], center = means, scale = sds)
numerocont <- which(colnames(prueba_dumm)%in%continuas)
prueba2 <-cbind(prueba2, prueba_dumm[,-numerocont])

library(e1071)
library(stats)
predicciones_precios_gasolSVM <- predict(modelo, newdata = prueba2) #estandarizadas

#Deshacemos la estandarización
```

```
predicciones_precios_gasolSVM1 <- (predicciones_precios_gasolSVM*0.1793881) + 1.178630
#(prediccion*sds)+mean

###Dividimos en 4 grupos las predicciones de reg lineal
q1_predic_svm = sample.split(predicciones_precios_gasolSVM1, SplitRatio = 0.25) #cuartil 1
q1_predic_svm = subset(precios_gasol_def_prueba_def, q1_predic_svm == TRUE)

q2_predic_svm = sample.split(predicciones_precios_gasolSVM1, SplitRatio = 0.5) #cuartil 2
q2_predic_svm = subset(precios_gasol_def_prueba_def, q2_predic_svm == TRUE)

q3_predic_svm = sample.split(predicciones_precios_gasolSVM1, SplitRatio = 0.75) #cuartil 3
q3_predic_svm = subset(precios_gasol_def_prueba_def, q3_predic_svm == TRUE)

q4_predic_svm <- sample(1:nrow(precios_gasol_def_prueba_def), size=303568,
replace=TRUE) #cuartil 4
q4_predic_svm <- precios_gasol_def_prueba_def[q4_predic_svm, ]

###Descriptivos en cada grupo/cuartil de las variables indep. usadas en svm:
#q1
min(q1_predic_svm$latitud) #27.75194
max(q1_predic_svm$latitud) #43.68742
mean(q1_predic_svm$latitud) #39.68211

min(q1_predic_svm$longitud) #-18.01194
max(q1_predic_svm$longitud) #6.390493
mean(q1_predic_svm$longitud) #-3.23847

table(q1_predic_svm$horario)

table(q1_predic_svm$rotulo)

table(q1_predic_svm$provincia)
write.table(table(q1_predic_svm$provincia),
file="C:/Users/Beatriz/Desktop/TFM_def/frecuencias_provincia_q1",
          sep=";", quote = FALSE, row.names=FALSE, dec = ",")

table(q1_predic_svm$tipo_gasol)
write.table(table(q1_predic_svm$tipo_gasol),
file="C:/Users/Beatriz/Desktop/TFM_def/frecuencias_tipo_gasol_q1",
          sep=";", quote = FALSE, row.names=FALSE, dec = ",")



#q2
min(q2_predic_svm$latitud) #27.75194
max(q2_predic_svm$latitud) #43.68742
mean(q2_predic_svm$latitud) #39.68095
min(q2_predic_svm$longitud) #-18.01194
max(q2_predic_svm$longitud) #6.390493
mean(q2_predic_svm$longitud) #-3.234914

table(q2_predic_svm$horario)
```

```
table(q2_predic_svm$rotulo)

table(q2_predic_svm$provincia)
write.table(table(q2_predic_svm$provincia),
file="C:/Users/Beatriz/Desktop/TFM_def/frecuencias_provincia_q2",
          sep=";", quote = FALSE, row.names=FALSE, dec = ",")

table(q2_predic_svm$tipo_gasol)


#q3
min(q3_predic_svm$latitud) #27.75194
max(q3_predic_svm$latitud) #43.68742
mean(q3_predic_svm$latitud) #39.68253

min(q3_predic_svm$longitud) #-18.01194
max(q3_predic_svm$longitud) #6.390493
mean(q3_predic_svm$longitud) #-3.231715

table(q3_predic_svm$horario)

table(q3_predic_svm$rotulo)

table(q3_predic_svm$provincia)
write.table(table(q3_predic_svm$provincia),
file="C:/Users/Beatriz/Desktop/TFM_def/frecuencias_provincia_q3",
          sep=";", quote = FALSE, row.names=FALSE, dec = ",")

table(q1_predic_svm$tipo_gasol)


#q4
min(q4_predic_svm$latitud) #27.75194
max(q4_predic_svm$latitud) #43.68742
mean(q4_predic_svm$latitud) #39.68238

min(q4_predic_svm$longitud) #-18.01194
max(q4_predic_svm$longitud) #6.390493
mean(q4_predic_svm$longitud) #-3.233438

table(q4_predic_svm$horario)

table(q4_predic_svm$rotulo)

table(q4_predic_svm$provincia)
write.table(table(q4_predic_svm$provincia),
file="C:/Users/Beatriz/Desktop/TFM_def/frecuencias_provincia_q4",
          sep=";", quote = FALSE, row.names=FALSE, dec = ",")

table(q4_predic_svm$tipo_gasol)
```

```
#############################################################################
#Predicciones
#############################################################################
sample1 = sample.split(precios_gasol_def, SplitRatio = 0.3)
precios_gasol_def_prueba = subset(precios_gasol_def, sample1 == TRUE) #predicción
write.table(precios_gasol_def_prueba,file="C:/Users/Beatriz/Desktop/TFM_def/precios_gasol_
def_prueba.csv", sep=";", quote = FALSE,
          row.names=FALSE, dec = ",")

#Creamos un conjunto de datos sin las variables que no foramn parte del modelo y sin la
variable predictor:
precios_gasol_def_prueba_def <- data.frame(precios_gasol_def_prueba[, c(-1, -2, -7, -9, -12)])

#Preparación del conjunto de datos (tiene que estar en la misma escala que el conjunto de datos
del modelo):
#a) EXtraemos variables continuas y categóricas
continuas = c("latitud", "longitud")
categor = c("horario","margen","rotulo","provincia", "tipo_gasol")
#Extraemos variables continuas sin var.objetivo
continuasin <- c("latitud", "longitud")
prueba <- precios_gasol_def_prueba_def[,c(continuas,categor)]
#b) Creamos dummys a las variables categóricas
prueba$horario = sapply(as.character(prueba$horario), switch, "no abre las 24 horas" = 1,
"abierto las 24 horas" = 2,
                        USE.NAMES = F)
levels(prueba$margen) <- relevel(prueba$margen, ref = "D")
levels(prueba$rotulo) <- relevel(prueba$rotulo, ref = "otros")
levels(prueba$provincia) <- relevel(prueba$provincia, ref = "BARCELONA")
levels(prueba$tipo_gasol) <- relevel(prueba$tipo_gasol, ref = "gasolina95Proteccion")

require(reshape2)
prueba_dumm$id <- rownames(prueba_dumm)
melt(prueba_dumm)

prueba_dumm <- data.frame(prueba$latitud, prueba$longitud, prueba$media_precio,
as.numeric(prueba$horario),
                          as.numeric(prueba$margen), as.numeric(prueba$rotulo),
                          as.numeric(prueba$provincia), as.numeric(prueba$tipo_gasol))
colnames(prueba_dumm) <- c("latitud","longitud",
"media_precio","horario","margen","rotulo","provincia",
                           "tipo_gasol")
prueba_dumm <- prueba_dumm[,c(-3, -10)]
#c)Estandarizamos las variables continuas, excepto la dependiente
#Calculamos medias y dtipica de datos y estandarizamos (solo las continuas)
means <- apply(prueba_dumm[,continuas],2,mean)
sds <- sapply(prueba_dumm[,continuas],sd)
#Estandarizamos las continuas y uno con las categóricas
prueba2 <- scale(prueba_dumm[,continuas], center = means, scale = sds)
numerocont <- which(colnames(prueba_dumm)%in%continuas)
prueba2 <-cbind(prueba2, prueba_dumm[,-numerocont])
```

```
library(e1071)
library(stats)
predicciones_precios_gasol_svm <- predict(svmlineal, newdata = prueba2)

#Vamos a dibujar las predicciones obtenidas y los datos originales, media_precio:
plot(precios_gasol_def_prueba_def2$media_precio, abs(predicciones_precios_gasol_svm),
col='blue',main='Gráfico de los datos frente a las
    predicciones SVM', xlim = c(0,3), xlab = "Precio medio",
    ylab = "Predicción SVM")
legend('bottomright',legend='Predicciones SVM',pch=18, col='blue', bty='n')

################################################################################
#Competencia empresarial/teoría de juegos
################################################################################
#La Gomera
library(ggmap)
library(ggplot2)
Madrid = get_map("La Gomera, Spain", zoom = 11)
p = ggmap(Madrid)
p + geom_point(data=gasolineras, aes(x=longitud, y=latitud), size=3)

#Precio de la gasolina de Las Canarias:
gasolineras_GranCan = which(gasolineras$provincia == "SANTA CRUZ DE TENERIFE" |
gasolineras$provincia == "PALMAS (LAS)")
gasolineras_GranCan = gasolineras[gasolineras_GranCan, c("direccion", "latitud", "longitud",
"localidad", "margen", "provincia",
                  "cp", "horario", "municipio", "rotulo")]
media_prGranCan = precios_gasol_def[
 which(
  is.element(
   precios_gasol_def[,"direccion"], gasolineras_GranCan[,"direccion"]
  ) &
   is.element(
    precios_gasol_def[,"latitud"], gasolineras_GranCan[,"latitud"]
   ) &
   is.element(
    precios_gasol_def[,"longitud"], gasolineras_GranCan[,"longitud"]
   ) &
   is.element(
    precios_gasol_def[,"localidad"], gasolineras_GranCan[,"localidad"]) &
   is.element(
    precios_gasol_def[,"margen"], gasolineras_GranCan[,"margen"]
   ) &
   is.element(
    precios_gasol_def[,"provincia"], gasolineras_GranCan[,"provincia"]
   ) &
   is.element(
    precios_gasol_def[,"cp"], gasolineras_GranCan[,"cp"]
   ) &
```

```
    is.element(
      precios_gasol_def[,"horario"], gasolineras_GranCan[,"horario"]
    ) &
    is.element(
      precios_gasol_def[,"municipio"], gasolineras_GranCan[,"municipio"]
    ) &
    is.element(
      precios_gasol_def[,"rotulo"], gasolineras_GranCan[,"rotulo"]
    )), c("media_precio")]

media_prGranCan

min(media_prGranCan)
max(media_prGranCan)


##Gráfico de sensibilidad
#Para distintos valores de alfa;
p1 = c(1.2, 1.1854, 1.1702, 1.1552, 1.1405, 1.1263, 1.1125, 1.0992, 1.0864, 1.0742, 1.0543,
1.0108, 0.96895, 0.93286,
    0.85238, 0.76038)

p2 = c(1.2, 1.1930, 1.1633, 1.1454, 1.128, 1.1114, 1.0956, 1.0806, 1.0664, 1.0529, 1.0395,
0.98499, 0.94184, 0.90527,
    0.81624, 0.72771)

p3 = c(1.2, 1.1379, 1.0884, 1.0479, 1.0142, 0.98562, 0.96101, 0.93958, 0.92073, 0.90400,
0.88877,
    0.83285, 0.79566, 0.76408, 0.78315, 0.73064)

p4 = c(1.2, 1.1379, 1.0884, 1.0479, 1.0142, 0.98562, 0.96101, 0.93958, 0.92073, 0.90400,
0.88877, 0.83285, 0.79566,
    0.76408, 0.78315, 0.73064)

alfa = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 2, 2.5, 3, 5)

sensibility = as.data.frame(cbind(p1, p2, p3, p4, alfa))
p = ggplot(sensibility, aes(alfa)) +
  geom_line(aes(y = p1, color="p1"), size=2) +
  geom_line(aes(y = p2, color="p2"), size=2) + geom_line(aes(y = p3, color="p3, p4"), size=2)
+
  labs(color = "Precio") + xlab("Alfa") + ylab("Precio (€)") + ggtitle("Gráfico de sensiblidad")
+
  theme(plot.title = element_text(hjust = 0.5))
p
```